User Manual

Arista Networks

www.aristanetworks.com

Arista EOS version 4.10.3
22 October 2012

Headquarters	Support	Sales
5470 Great America Parkway Santa Clara, CA 95054 USA		
408 547-5500	408 547-5502 866 476-0000	408 547-5501 866 497-0000
www.aristanetworks.com	support@aristanetworks.com	sales@aristanetworks.com

© Copyright 2012 Arista Networks, Inc. The information contained herein is subject to change without notice. Arista Networks and the Arista logo are trademarks of Arista Networks, Inc in the United States and other countries. Other product or service names may be trademarks or service marks of others.

Table of Contents

Table of Contents 3

Command Reference 9

Preface 27
Audience 27
Organization 27

Chapter 1 Product Overview 29
Supported Features 29
Feature Availability on Switch Platforms 32

Chapter 2 Initial Configuration and Recovery 35
Initial Switch Access 35
Connection Management 40
Recovery Procedures 41
Upgrades 45
Session Management Commands 47

Chapter 3 Command-Line Interface 53
Accessing the EOS CLI 53
Processing Commands 54
Switch Platforms 59
Command Modes 60
Managing Switch Configuration Settings 63
Other Command-Line Interfaces 66
Directory Structure 67
Command-Line Interface Commands 68

Chapter 4 AAA Configuration 95
Authorization, Authentication, and Accounting Overview 95
Configuring the Security Services 96
Activating Security Services 104
Security Configuration Examples 107
AAA Commands 109

Chapter 5 Administering the Switch. 151
Managing the Switch Name 151
Managing the System Clock 153
Managing Display Attributes 155
Event Monitor 157
Switch Administration Commands 160

Chapter 6 Booting the Switch. 197
Boot Loader – Aboot 197
Configuration Files 198
System Reset 202
Aboot Shell. 208
Aboot Configuration Commands 212
Switch Booting Commands 217

Chapter 7 Switch Environment Control 233
Environment Control Introduction 233
Environment Control Overview 233
Configuring and Viewing Environment Settings 235
Environment Commands 239

Chapter 8 Ethernet Ports 249
Ethernet Ports Introduction 249
Ethernet Standards. 249
Ethernet Physical Layer. 251
Interfaces 253
Ethernet Configuration Procedures 254
Ethernet Configuration Commands 262

Chapter 9 Port Channels and LACP. 287
Port Channel Introduction 287
Port Channel Conceptual Overview. 287
Configuration Procedures. 289
Port Channel and LACP Configuration Commands 293

Chapter 10 Data Transfer 323
Introduction 323
Data Transfer Methods 324
Ports 327
Routing Concepts 330
Data Transfer Configuration Processes 331
Data Transfer Command Descriptions 340

Chapter 11 VLANs 385
Introduction 385
VLAN Conceptual Overview 385
VLAN Configuration Procedures 388
VLAN Configuration Commands 394

Chapter 12 Multi-Chassis Link Aggregation 423
MLAG Introduction 423
MLAG Conceptual Overview. 424
MLAG Maintenance. 426
Configuring MLAG 429
MLAG Implementation Example. 434
MLAG Commands 442

Chapter 13 ACLs and Route Maps 455
ACL, Route Map, and Prefix List Introduction 455
Access Control Lists 456
Route Maps 470
Prefix Lists 476
ACL, Route Map, and Prefix List Commands 479

Chapter 14 VRRP and VARP 531
VRRP and VARP Conceptual Overview 531
VRRP and VARP Implementation Procedures 533
VRRP and VARP Implementation Examples. 537
VRRP and VARP Configuration Commands. 542

Chapter 15 Spanning Tree Protocol. 561
Introduction to Spanning Tree Protocols 561
Spanning Tree Overview 561
Configuring a Spanning Tree 568
STP Commands 581

Chapter 16 Quality of Service. 637
Quality of Service Conceptual Overview. 637
Quality of Service Configuration Procedures 646
Quality of Service Configuration Commands 652

Chapter 17 IPv4 681
Introduction 681
IPv4 Description 681
Configuring IPv4 683

IPv4 Command Descriptions 689

Chapter 18 IPv6 727
Introduction 727
IPv6 Description 727
Configuring IPv6 730
IPv6 Command Descriptions 736

Chapter 19 Open Shortest Path First – Version 2 763
OSPFv2 Introduction 763
OSPFv2 Conceptual Overview 764
Configuring OSPFv2 767
OSPFv2 Examples. 781
OSPFv2 Commands 790

Chapter 20 Open Shortest Path First – Version 3 837
OSPFv3 Introduction 837
OSPFv3 Conceptual Overview 838
Configuring OSPFv3 841
OSPFv3 Examples. 851
OSPFv3 Commands 860

Chapter 21 Border Gateway Protocol 889
BGP Conceptual Overview. 889
Configuring BGP 891
BGP Examples 901
BGP Commands 904

Chapter 22 Routing Information Protocol 975
RIP Conceptual Overview 975
Running RIP on the Switch 976
RIP Commands 979

Chapter 23 Multicast 991
Introduction 991
Multicast Architecture 993
Multicast Protocols 995
Configuring Multicast 998
Multicast Example 1008
Multicast Commands 1012
IGMP Commands. 1024
IGMP Snooping Commands 1042

PIM Commands 1075

Chapter 24 SNMP 1105
SNMP Introduction 1105
SNMP Conceptual Overview 1105
Configuring SNMP 1108
SNMP Commands 1116

Chapter 25 Latency Analyzer (LANZ) 1145
Introduction to LANZ 1145
LANZ Overview 1145
Configuring LANZ 1146
LANZ Commands 1153

Chapter 26 VM Tracer 1165
VM Tracer Introduction 1165
VM Tracer Conceptual Overview 1165
VM Tracer Configuration Procedures 1166
VM Tracer Configuration Commands 1170

Chapter 27 sFlow 1183
sFlow Conceptual Overview 1183
sFlow Configuration Procedures 1186
SFlow Configuration Commands 1188

Glossary 1201

Index 1205

Command Reference

Preface 27

Chapter 1 Product Overview. 29

Chapter 2 Initial Configuration and Recovery 35
idle-timeout 48
management 49
show inventory 50
shutdown (Management-Telnet) 51

Chapter 3 Command-Line Interface 53
action bash 69
alias 70
bash 71
comment (various configuration modes) 72
configure (configure terminal) 73
configure network. 74
copy running-config 75
daemon 76
delay 77
enable 78
end 79
event-handler 80
exit (Global Configuration) 81
ip ftp client source-interface 82
ip http client source-interface 83
ip ssh client source-interface 84
ip tftp client source-interface. 85
schedule 86
show (various configuration modes) 87
show event-handler 88
show schedule 89
show schedule summary 90
terminal length 91
terminal monitor 92
trigger 93

Chapter 4 AAA Configuration 95
aaa accounting 110
aaa authentication enable 111
aaa authentication login 112
aaa authentication policy local 113
aaa authorization commands 114
aaa authorization config-commands 115
aaa authorization console 116
aaa authorization exec 117
aaa group server radius 118
aaa group server tacacs+ 119
aaa root 120
clear aaa counters 121
clear aaa counters radius 122
clear aaa counters tacacs+ 123
enable secret 124
ip radius source-interface 125
ip tacacs source-interface 126
radius-server deadtime 127
radius-server host 128
radius-server key 130
radius-server retransmit 131
radius-server timeout 132
server (server-group-RADIUS configuration mode) 133
server (server-group-TACACS+ configuration mode) 134
show aaa 135
show aaa counters 136
show aaa method-lists 137
show aaa sessions 138
show privilege 139
show radius 140
show tacacs 141
show user-account 142
show users 143
tacacs-server host 144
tacacs-server key 146
tacacs-server timeout 147
username 148
username sshkey 150

Chapter 5 Administering the Switch 151
banner login 161
banner motd 162
clock set 163
clock timezone 164
email 165
no event-monitor 166
event-monitor <log enable> 167

event-monitor backup max-size 168
event-monitor backup path 169
event-monitor buffer max-size 170
event-monitor clear 171
event-monitor interact 172
event-monitor sync 173
hostname 174
ip domain-name 175
ip host 176
ip name-server 177
ipv6 host 178
ntp bind 179
ntp server 180
ntp source 182
prompt 183
show banner 185
show clock 186
show event-monitor arp 187
show event-monitor mac 188
show event-monitor route 189
show event-monitor sqlite 190
show hosts 191
show ip domain-name 192
show ip name-server 193
show ntp associations 194
show ntp status 195

Chapter 6 Booting the Switch 197

Aboot Configuration Commands 212
CONSOLESPEED 213
NET commands 214
PASSWORD (ABOOT) 215
SWI 216

Switch Booting Commands 217
boot console 218
boot secret 219
boot system 221
protocol 222
redundancy 223
redundancy force-switchover 224
reload 225
reload <scheduled> 227
show redundancy file-replication 228
show redundancy states 229
show redundancy switchover sso 230
show reload 231
show reload cause 232

Chapter 7 Switch Environment Control 233
environment fan-speed 240
environment insufficient-fans action. 241
environment overheat action 242
show environment all. 243
show environment cooling 244
show environment power 245
show environment temperature. 246

Chapter 8 Ethernet Ports 249
flowcontrol receive 263
flowcontrol send 264
hardware port-group 265
interface ethernet 267
interface management 268
link-debounce 269
mac-address 270
show flowcontrol. 271
show hardware port-group 272
show interfaces capabilities 273
show interfaces counters 274
show interfaces counters bins 275
show interfaces counters errors 276
show interfaces counters queue 277
show interfaces counters rates 278
show interfaces negotiation. 279
show interfaces phy 280
show interfaces status. 282
show interfaces status errdisabled 283
show interfaces transceiver 284
show interfaces transceiver properties 285
speed 286

Chapter 9 Port Channels and LACP. 287
channel-group 294
interface port-channel 296
lacp port-priority 297
lacp rate. 298
lacp system-priority 299
port-channel lacp fallback 300
port-channel lacp fallback timeout. 301
port-channel load-balance 302
port-channel load-balance fields 303
port-channel min-links. 307
show etherchannel 308
show lacp aggregates 309
show lacp counters 310

show lacp interface 311
show lacp internal 313
show lacp neighbor 314
show lacp sys-id 316
show port-channel 317
show port-channel limits 319
show port-channel load-balance fields 320
show port-channel summary 321
show port-channel traffic 322

Chapter 10 Data Transfer 323
clear mac address-table dynamic 342
control-plane 343
description 344
exclude 345
interface loopback 346
ip access-group (Control Plane mode) 347
ip distributed-hosts 348
mac address-table aging-time 349
mac address-table static 350
monitor session destination 353
monitor session source 354
mtu 355
no monitor session 356
rd (VRF configuration mode) 357
routing-context vrf 358
show interfaces description 359
show mac address-table 360
show mac address-table aging time 362
show mac address-table count 363
show monitor session 364
show port-security 365
show port-security address 366
show port-security interface 367
show routing-context vrf 368
show storm-control 369
show switch forwarding-mode 370
show track 371
show vrf 372
storm-control 373
switch forwarding-mode 375
switchport 376
switchport mac address learning 377
switchport port-security 378
switchport port-security maximum 379
threshold fecs 380
track 381
vlan (distributed-hosts configuration) 382

vrf definition 383
vrf forwarding 384

Chapter 11 VLANs 385
autostate 395
interface vlan 396
name (VLAN configuration mode) 397
private-vlan 398
private-vlan mapping 399
show dot1q-tunnel 400
show interfaces switchport 401
show interfaces switchport backup 402
show interfaces trunk 403
show interfaces vlans 404
show vlan 405
show vlan dynamic 406
show vlan internal allocation policy 407
show vlan internal usage 408
show vlan private-vlan 409
show vlan summary 410
show vlan trunk group 411
state 412
switchport access vlan 413
switchport mode 414
switchport private-vlan mapping 415
switchport trunk allowed vlan 416
switchport trunk group 417
switchport trunk native vlan 418
trunk group 419
vlan 420
vlan internal allocation policy 421

Chapter 12 Multi-Chassis Link Aggregation 423
domain-id 443
heartbeat-interval 444
local-interface 445
mlag (port-channel interface configuration) 446
mlag configuration (global configuration) 447
peer-address 448
peer-link 449
reload-delay 450
show mlag 451
show mlag interfaces 453
shutdown (MLAG) 454

Chapter 13 ACLs and Route Maps 455
clear ip access-lists counters 481
clear ipv6 access-lists counters 482
continue (route-map configuration mode) 483
deny (IPv4 ACL) 484
deny (IPv6 ACL) 486
deny (IPv6 Prefix List) 488
deny (MAC ACL) 489
deny (Standard IPv4 ACL) 491
deny (Standard IPv6 ACL) 492
ip access-group (Interface Configuration modes) 493
ip access-list 494
ip access-list standard 495
ip prefix-list 496
ipv6 access-group 497
ipv6 access-list 498
ipv6 access-list standard 499
ipv6 prefix-list 500
mac access-group 501
mac access-list 502
match (route-map configuration mode) 503
no <sequence number> 504
permit (IPv4 ACL) 505
permit (IPv6 ACL) 507
permit (IPv6 Prefix List) 509
permit (MAC ACL) 510
permit (Standard IPv4 ACL) 512
permit (Standard IPv6 ACL) 513
remark 514
resequence 515
route-map 516
seq (IPv6 Prefix Lists) 518
set (route-map configuration mode) 519
show (ACL configuration modes) 520
show (route-map configuration mode) 522
show ip access-lists 523
show ip prefix-list 524
show ipv6 access-lists 525
show ipv6 prefix-list 526
show mac access-lists 527
show route-map 528
statistics per-entry (ACL configuration modes) 529

Chapter 14 VRRP and VARP 531
ip virtual-router address 543
ip virtual-router mac-address 544
ip virtual-router mac-address advertisement-interval 545
no vrrp 546

show ip virtual-router 547
show vrrp 548
vrrp authentication 550
vrrp description 551
vrrp ip 552
vrrp ip secondary 553
vrrp preempt 554
vrrp preempt delay 555
vrrp priority 557
vrrp shutdown 558
vrrp timers advertise 559
vrrp track 560

Chapter 15 Spanning Tree Protocol 561
abort (mst-configuration mode) 583
clear spanning-tree counters 584
clear spanning-tree counters session 585
clear spanning-tree detected-protocols 586
exit (mst-configuration mode) 587
instance 588
name (mst-configuration mode) 589
revision (mst-configuration mode) 590
show (mst-configuration mode) 591
show spanning-tree 592
show spanning-tree blockedports 595
show spanning-tree bridge 596
show spanning-tree counters 597
show spanning-tree interface 598
show spanning-tree mst 599
show spanning-tree mst configuration 601
show spanning-tree mst interface 602
show spanning-tree mst test information 603
show spanning-tree root 604
show spanning-tree topology status 605
spanning-tree bpdufilter 606
spanning-tree bpduguard 607
spanning-tree bpduguard rate-limit count (global) 608
spanning-tree bpduguard rate-limit count (interface) 609
spanning-tree bpduguard rate-limit default 610
spanning-tree bpduguard rate-limit enable / disable 611
spanning-tree bridge assurance 612
spanning-tree cost 613
spanning-tree forward-time 614
spanning-tree guard 615
spanning-tree hello-time 616
spanning-tree link-type 617
spanning-tree loopguard default 618
spanning-tree max-age 619

spanning-tree max-hops 620
spanning-tree mode 621
spanning-tree mst configuration 622
spanning-tree portchannel guard misconfig 623
spanning-tree portfast 624
spanning-tree portfast auto 625
spanning-tree portfast bpdufilter default 626
spanning-tree portfast bpduguard default 627
spanning-tree portfast <port type> 628
spanning-tree port-priority 629
spanning-tree priority 630
spanning-tree root 631
spanning-tree transmit hold-count 632
spanning-tree vlan 633
switchport backup interface 634

Chapter 16 Quality of Service 637
bandwidth percent (Transmit Queue Configuration Modes) 653
mc-tx-queue 655
platform petraA traffic-class 656
priority (FM4000 and Petra platform switches) 657
priority (Trident platform switches) 659
qos cos 661
qos dscp 662
qos trust 663
qos map cos 664
qos map dscp 665
qos map traffic-class to cos 666
qos map traffic-class to dscp 667
qos map traffic-class to mc-tx-queue 668
qos map traffic-class to tx-queue 670
qos map traffic-class to uc-tx-queue 671
qos rewrite cos 673
qos rewrite dscp 674
shape rate (Interface configuration mode) 675
shape rate (Transmit Queue configuration modes) 676
show qos interface 677
show qos maps 678
tx-queue 679
uc-tx-queue 680

Chapter 17 IPv4 681
arp 690
arp timeout 691
clear arp-cache 692
clear ip arp 693
clear ip dhcp relay counters 694

ip address . . . 696
ip dhcp relay . . . 697
ip dhcp relay always-on . . . 698
ip dhcp relay circuit-id . . . 699
ip dhcp relay client server . . . 700
ip dhcp relay information option (Global) . . . 701
ip dhcp relay information option circuit-id . . . 702
ip dhcp relay log verbose . . . 703
ip dhcp relay server . . . 704
ip helper-address . . . 705
ip load-sharing . . . 706
ip local-proxy-arp . . . 707
ip proxy-arp . . . 708
ip route . . . 709
ip routing . . . 711
show arp . . . 712
show ip . . . 713
show ip arp . . . 714
show ip dhcp relay . . . 716
show ip dhcp relay counters . . . 717
show ip helper-address . . . 718
show ip interface . . . 719
show ip interface brief . . . 720
show ip route . . . 721
show ip route gateway . . . 723
show ip route host . . . 724
show ip route summary . . . 725
show ip route tag . . . 726

Chapter 18 IPv6 . . . 727
clear ipv6 neighbors . . . 737
ipv6 address . . . 738
ipv6 enable . . . 739
ipv6 hardware fib aggregate-address . . . 740
ipv6 nd managed-config-flag . . . 741
ipv6 nd other-config-flag . . . 742
ipv6 nd prefix . . . 743
ipv6 nd ra interval . . . 745
ipv6 nd ra lifetime . . . 746
ipv6 nd ra mtu suppress . . . 747
ipv6 nd ra suppress . . . 748
ipv6 nd reachable-time . . . 749
ipv6 nd router-preference . . . 750
ipv6 neighbor . . . 751
ipv6 route . . . 752
ipv6 unicast-routing . . . 753
show ipv6 hardware fib aggregate-address . . . 754
show ipv6 interface . . . 755

show ipv6 nd ra internal state 756
show ipv6 neighbor 757
show ipv6 route 758
show ipv6 route host 759
show ipv6 route interface 760
show ipv6 route tag 761

Chapter 19 Open Shortest Path First – Version 2 763
area (OSPFv2) 791
area default-cost (OSPFv2) 793
area filter (OSPFv2) 794
area range (OSPFv2) 795
auto-cost reference-bandwidth (OSPFv2) 796
distance ospf intra-area (OSPFv2) 797
exit (OSPFv2) 798
ip ospf authentication 799
ip ospf authentication-key 800
ip ospf cost 801
ip ospf dead-interval 802
ip ospf hello-interval 803
ip ospf message-digest-key 804
ip ospf name-lookup 805
ip ospf network 806
ip ospf priority 807
ip ospf retransmit-interval 808
ip ospf shutdown 809
ip ospf transmit-delay 810
log-adjacency-changes (OSPFv2) 811
max-lsa (OSPFv2) 812
maximum-paths (OSPFv2) 813
network area (OSPFv2) 814
passive-interface default (OSPFv2) 815
passive-interface <interface> (OSPFv2) 816
point-to-point routes (OSPFv2) 817
redistribute (OSPFv2) 818
router-id (OSPFv2) 819
router ospf 820
show ip ospf 821
show ip ospf border-routers 822
show ip ospf database database-summary 823
show ip ospf database <link state list> 824
show ip ospf database <link-state details> 825
show ip ospf interface 828
show ip ospf interface brief 829
show ip ospf neighbor 830
show ip ospf request-list 832
show ip ospf retransmission-list 833
shutdown (OSPFv2) 834

timers spf (OSPFv2) 835

Chapter 20 Open Shortest Path First – Version 3 837
area (OSPFv3) 861
area default-cost (OSPFv3) 862
clear ipv6 ospf force-spf 863
default-metric (OSPFv3) 864
distance ospf intra-area (OSPFv3) 865
exit (OSPFv3) 866
ipv6 ospf area 867
ipv6 ospf cost 868
ipv6 ospf dead-interval 869
ipv6 ospf hello-interval 870
ipv6 ospf network 871
ipv6 ospf priority 872
ipv6 ospf retransmit-interval 873
ipv6 ospf transmit-delay 874
ipv6 router ospf 875
log-adjacency-changes (OSPFv3) 876
maximum-paths (OSPFv3) 877
passive-interface (OSPFv3) 878
redistribute (OSPFv3) 879
router-id (OSPFv3) 880
show ipv6 ospf 881
show ipv6 ospf border-routers 882
show ipv6 ospf database 883
show ipv6 ospf interface 884
show ipv6 ospf neighbor 885
shutdown (OSPFv3) 886
timers spf (OSPFv3) 887

Chapter 21 Border Gateway Protocol 889
address-family 906
aggregate-address 907
bgp client-to-client reflection 909
bgp cluster-id 910
bgp default 911
bgp listen limit 913
bgp listen range 914
bgp log-neighbor-changes 915
clear ip bgp 916
clear ipv6 bgp 917
distance bgp 918
ip as-path access-list 919
ip community-list expanded 920
ip community-list standard 921
ip extcommunity-list expanded 922

ip extcommunity-list standard 923
maximum paths (BGP) 924
neighbor activate 925
neighbor allowas-in 926
neighbor description 927
neighbor ebgp-multihop 928
neighbor export-localpref 929
neighbor import-localpref 930
neighbor local-as 931
neighbor local-v6-addr 932
neighbor maximum-routes 933
neighbor next-hop-peer 934
neighbor next-hop-self 935
neighbor out-delay 936
neighbor password 937
neighbor peer-group (create) 938
neighbor peer-group (neighbor assignment) 940
neighbor remote-as 941
neighbor remove-private-as 942
neighbor route-map (BGP) 943
neighbor route-reflector-client 944
neighbor send-community 945
neighbor shutdown 946
neighbor soft-reconfiguration 947
neighbor timers 948
neighbor update-source 949
neighbor weight 950
network (BGP) 951
no neighbor 952
redistribute (BGP) 953
router-id (BGP) 954
router bgp 955
show ip as-path access-list 956
show ip bgp 957
show ip bgp community 958
show ip bgp neighbors 959
show ip bgp neighbors (route type) 960
show ip bgp neighbors (route-type) community 961
show ip bgp paths 962
show ip bgp peer-group 963
show ip bgp summary 964
show ip community-list 965
show ip extcommunity-list 966
show ipv6 bgp 967
show ipv6 bgp community 968
show ipv6 bgp neighbors 969
show ipv6 bgp neighbors (route type) 970
show ipv6 bgp neighbors (route type) community 971
show ipv6 bgp summary 972
shutdown (BGP) 973

timers bgp 974

Chapter 22 Routing Information Protocol 975
default-metric 980
distance (RIP) 981
ip rip v2-broadcast 982
network (RIP) 983
redistribute (RIP) 984
router rip 985
show ip rip database 986
show ip rip neighbors 987
shutdown (RIP) 988
timers basic (RIP) 989

Chapter 23 Multicast 991

Multicast Commands 1012
clear ip mfib fastdrop 1013
clear ip mroute 1014
ip mfib activity polling-interval 1015
ip mfib fastdrop 1016
ip mfib max-fastdrops 1017
ip multicast boundary 1018
ip multicast-routing 1019
show ip mfib 1020
show ip mroute 1022
show ip mroute count 1023

IGMP Commands 1024
clear ip igmp group 1025
ip igmp last-member-query-count 1026
ip igmp last-member-query-interval 1027
ip igmp query-interval 1028
ip igmp query-max-response-time 1029
ip igmp startup-query-count 1030
ip igmp startup-query-interval 1031
ip igmp static-group 1032
ip igmp static-group acl 1033
ip igmp static-group range 1034
ip igmp version 1036
show ip igmp groups 1037
show ip igmp interface 1038
show ip igmp static-groups 1039
show ip igmp static-groups acl 1040
show ip igmp static-groups group 1041

IGMP Snooping Commands 1042
clear ip igmp snooping counters 1043

ip igmp profile 1044
ip igmp snooping 1046
ip igmp snooping filter 1047
ip igmp snooping immediate-leave 1048
ip igmp snooping querier 1049
ip igmp snooping querier address 1050
ip igmp snooping querier max-response-time 1051
ip igmp snooping querier query-interval 1052
ip igmp snooping robustness-variable 1053
ip igmp snooping vlan 1054
ip igmp snooping vlan max-groups 1055
ip igmp snooping vlan mrouter 1056
ip igmp snooping vlan querier 1057
ip igmp snooping vlan querier address 1059
ip igmp snooping vlan querier max-response-time 1060
ip igmp snooping vlan querier query-interval 1061
ip igmp snooping vlan static 1062
permit / deny 1063
range 1064
show ip igmp profile 1065
show ip igmp snooping 1066
show ip igmp snooping counters 1067
show ip igmp snooping counters ethdev-pams 1068
show ip igmp snooping groups 1069
show ip igmp snooping groups count 1072
show ip igmp snooping mrouter 1073
show ip igmp snooping querier 1074

PIM Commands 1075

ip pim anycast-rp 1076
ip pim border-router 1077
ip pim bsr-border 1078
ip pim bsr-candidate 1079
ip pim bsr-holdtime 1080
ip pim dr-priority 1081
ip pim join-prune-interval 1082
ip pim log-neighbor-changes 1083
ip pim neighbor-filter 1084
ip pim query-interval 1085
ip pim register-source 1086
ip pim rp-address 1087
ip pim rp-candidate 1088
ip pim sparse-mode 1090
ip pim sparse-mode sg-expiry-timer 1091
ip pim spt-threshold 1092
ip pim ssm range 1093
show ip pim bsr 1094
show ip pim config-sanity 1095
show ip pim interface 1096
show ip pim neighbor 1097

show ip pim protocol 1098
show ip pim register-source 1099
show ip pim rp. 1100
show ip pim rp-candidate 1101
show ip pim rp-hash. 1102
show ip pim upstream joins 1103

Chapter 24 SNMP 1105
no snmp-server 1117
show snmp 1118
show snmp chassis 1119
show snmp community 1120
show snmp contact 1121
show snmp engineID 1122
show snmp group 1123
show snmp host 1124
show snmp location 1125
show snmp mib 1126
show snmp user 1127
show snmp view 1128
snmp-server chassis-id 1129
snmp-server community 1130
snmp-server contact 1131
snmp-server enable traps. 1132
snmp-server engineID local 1133
snmp-server engineID remote 1134
snmp-server extension 1135
snmp-server group 1136
snmp-server host. 1137
snmp-server location 1139
snmp-server source-interface 1140
snmp-server user. 1141
snmp-server view 1142
snmp-server vrf 1143
snmp trap link-status 1144

Chapter 25 Latency Analyzer (LANZ) 1145
max-connections 1154
queue-monitor length (global configuration mode) 1155
queue-monitor length (interface configuration mode). 1156
queue-monitor length log 1158
queue-monitor streaming 1159
show queue-monitor length 1160
show queue-monitor length csv 1161
show queue-monitor length status 1162
shutdown (queue-monitor-streaming configuration) 1163

Chapter 26 VM Tracer 1165
allowed-vlan 1171
autovlan disable. 1172
exit (vmtracer mode). 1173
password (vmtracer mode) 1174
show vmtracer interface. 1175
show vmtracer session 1176
show vmtracer vm 1177
url. 1178
username (vmtracer mode) 1179
vmtracer 1180
vmtracer session 1181

Chapter 27 sFlow 1183
clear sflow counters 1189
sflow destination 1190
sflow enable 1191
sflow polling-interval 1192
sflow run. 1193
sflow sample 1194
sflow source 1195
sflow source-interface. 1196
show sflow 1197
show sflow interfaces 1199

Preface

This preface describes who should read this document and how it is organized.

Audience

This guide is for experienced network administrators who are responsible for configuring and maintaining Arista Switches.

Organization

This manual is organized into the following chapters:

Chapter	Title	Description
Chapter 1	Product Overview	Presents an overview of the Arista EOS software for the 7100 series switches.
Chapter 2	Initial Configuration and Recovery	Describes initial configuration and switch recovery tasks.
Chapter 3	Command-Line Interface	Describes how to use the CLI.
Chapter 4	AAA Configuration	Describes use of the local database, TACACS+ servers, and RADIUS servers to authenticate users and authorize tasks.
Chapter 5	Administering the Switch	Describes administrative tasks, including clock maintenance and display options.
Chapter 6	Booting the Switch	Describes startup and upgrade procedures.
Chapter 7	Switch Environment Control	Describes commands that display temperature, fan, and power supply status.
Chapter 8	Ethernet Ports	Describes Ethernet ports supported by Arista switches.
Chapter 9	Port Channels and LACP	Describes port channel commands and configuration procedures.
Chapter 10	Data Transfer	Describes concepts and processes that are referenced by routing and switching protocols that Arista switches support.
Chapter 11	VLANs	Describes Arista's VLAN implementation, including private VLANs
Chapter 12	Multi-Chassis Link Aggregation	An MLAG is a set of ports, residing on two cooperating switches, that appear to external devices as an ordinary link aggregation group.

Chapter	Title	Description
Chapter 13	ACLs and Route Maps	Describes the creation and implementation of rule-based lists, including ACLs, route maps, and prefix lists.
Chapter 14	VRRP and VARP	Describes Arista support of virtual IP addresses through the Virtual Router Redundancy Protocol and the Virtual-ARP feature.
Chapter 15	Spanning Tree Protocol	Spanning Tree Protocols prevent bridging loops in Layer 2 Ethernet networks.
Chapter 16	Quality of Service	Quality of Service defines a method of differentiating data streams to provide varying levels of service to the different streams.
Chapter 17	IPv4	Internet Protocol Version 4 defines a protocol for relaying packets across connected networks. Network devices are assigned 32 bit addresses.
Chapter 18	IPv6	Internet Protocol Version 6 defines a protocol for relaying packets across a set of connected networks. Each network device is assigned a 128 bit address.
Chapter 19	Open Shortest Path First – Version 2	Open Shortest Path First – Version 2 (OSPFv2) is a link-state routing protocol that operates over IPv4 within a single autonomous system.
Chapter 20	Open Shortest Path First – Version 3	Open Shortest Path First – Version 3 (OSPFv3) is a link-state routing protocol that operates over IPv6 within a single autonomous system.
Chapter 21	Border Gateway Protocol	Border Gateway Protocol (BGP) is an exterior gateway protocol (EGP) that exchanges routing information among neighboring routers in different Autonomous Systems (AS).
Chapter 22	Routing Information Protocol	Routing Information Protocol (RIP) is a distance-vector routing protocol typically used as an interior gateway protocol (IGP).
Chapter 23	Multicast	IP multicast is the transmission of data packets to a subset of all hosts. Arista switches support multicast transmissions through IGMP and PIM.
Chapter 24	SNMP	SNMP is an application-layer protocol that provides a standardized framework and a common language to monitor and manage network devices.
Chapter 25	Latency Analyzer (LANZ)	The Latency Analyzer (LANZ) is a family of EOS features that provide enhanced visibility into network dynamics, particularly in areas related to the delay packets experience through the network.
Chapter 26	VM Tracer	VM Tracer is a switch feature that determines the network configuration and requirements of connected VMWare hypervisors.
Chapter 27	sFlow	sFlow is a multi-vendor sampling technology that continuously monitors application level traffic flow at wire speed simultaneously on all interfaces.

Chapter 1

Product Overview

Arista switches feature high density, non-blocking 10 Gigabit Ethernet switches through an extensible modular network operating system.

This chapter provides an overview of features and summarizes the location of configuration and operational information. Topics covered by this chapter include:

- Supported Features
- Feature Availability on Switch Platforms

1.1 Supported Features

1.1.1 Management and Security Utilities

The following features configure, maintain, and secure the switch and its network connections:

- **Extensible Operating System (EOS)**: EOS is the interface between the switch and the software that controls the switch and manages the network. Refer to Section 3.1: Accessing the EOS CLI.
- **Linux Bash CLI**: The Bash shell accesses the underlying Linux operating system and extensions added through EOS. Refer to Section 3.6.2: Bash Shell.
- **DHCP Relay**: DHCP Relay is an agent that transmits Dynamic Host Configuration Protocol (DHCP) messages between clients and servers on different IP networks. Refer to Section 17.2.3: DHCP Relay.
- **Ethernet Management Port**s: Ethernet management Ports access the EOS management plane. Refer to Section 8.4.2: Management Interfaces.
- **Debugging Facilities**: The Bash shell includes utilities, such as traceroute and tcpdump, to maintain network extensions and diagnose connection issues.
- **Switch File Management**: File management facilitates adding, removing, and transferring switch files, including updated images. Refer to Section 3.7: Directory Structure.
- **Secure Shell**: Secure Shell provides secure login access to the switch from other network locations. Refer to Section 3.1: Accessing the EOS CLI.
- **Simple Network Management Protocol (SNMP)**: SNMP is a UDP-based network protocol that monitors network devices for error and alert conditions. Refer to Chapter 24, starting on page 1105.
- **Port Mirroring**: Port Mirroring sends a copy of network packets seen on one port to a network monitoring connection on a different port. Refer to Section 10.3.1, Port Mirroring, on page 327.

- **Virtual Router Redundancy Protocol (VRRP)**: VRRP increases network availability by defining a virtual router. Refer to Chapter 14, starting on page 531.
- **Control Plane Policing**: Control Plane Policing prioritizes control plane and management traffic and limits the rate of CPU bound control plane traffic to prevent denial of service traffic. Refer to Chapter 13, starting on page 433.
- **Authentication Services – Local, RADIUS, and TACACS+**: These services authenticate and authorize network users. Refer to Chapter 4, starting on page 95.
- **Access Control Lists (ACLs)**: ACLs filter network traffic. Refer to Chapter 13, starting on page 433.
- **Port Security**: Port Security limits the number of MAC addresses that can appear on a port. Refer to Section 10.3.6, Port Security, on page 329.
- **Storm Control**: Storm control terminates broadcast traffic forwarding when inbound broadcast frames consume excessive bandwidth. Refer to Section 10.3.2, Storm Control, on page 327.
- **In-Service-Software-Update (ISSU)**: In-Service-Software-Update updates switch software without disrupting packet forwarding. Refer to Section 2.4, Upgrades, on page 45.

1.1.2 Layer 2 Software Features

Arista switches support these layer 2 software features:

- **Link Aggregation:** The Link Aggregation Control Protocol (LACP), described by IEEE 802.3ad, defines a method for two switches to automatically establish and maintain LAGs. Link aggregation groups (LAGs) combine multiple ports in parallel to increase the link speed and provide higher availability. Refer to Chapter 9, starting on page 279.
- **Jumbo Frames**: Jumbo Frames are Ethernet frames with more than 1,500 bytes of payload. Refer to Section 10.3.8, Maximum Transmission Units (MTU), on page 329.
- **Link Layer Discovery Protocol (LLDP)**: LLDP advertises device identities, capabilities, and interconnections on local area networks.
- **Multi-Chassis Link Aggregation Protocol (MLAG)**: MLAG configures ports belonging to two cooperating switches such that they appear, to external devices, as an ordinary link aggregation group. Refer to Chapter 12, starting on page 423
- **Spanning Tree Protocols (STP)**: Spanning Tree Protocols are link layer network protocols that ensure a loop-free topology for any bridged LAN. Switches support these protocols:
  - — ***Rapid Spanning Tree Protocol (RSTP)***: Rapid Spanning Tree Protocol is an STP extension that provides faster convergence after a topology change.
  - — ***Multiple Spanning Tree Protocol (MSTP)***: MSTP is an RSTP extension that supports multiple VLAN groups.
  - — ***Per-VLAN Rapid Spanning Tree (PVRST+)***: Per-VRST+ is an RSTP extension that deploys a spanning tree for each VLAN.

  Refer to Chapter 15, starting on page 561.
- **Quality of Service (QoS)**: QoS prioritizes network traffic to guarantee dataflow performance levels. Supported QoS methods include:
  - — ***Priority Flow Control (PFC)***: PFC is a link level flow control mechanism that is independently controllable for each Class of Service (CoS).

— *Data Center Bridging Exchange (DCBX)*: DCBX is a discovery and capability exchange protocol that conveys configuration and attribute information between network devices to ensure consistent configuration across the network.

- **Virtual Local Area Networks (VLANs)**: VLANs define network device groups that communicate from the same broadcast domain, regardless of their physical location. VLANs are supported through these features:

  — *IEEE 802.1Q*: 802.1Q is a networking standard that allows multiple bridged networks to transparently share the same physical network link.

  — *IEEE 802.1ad*: 802.1ad is a networking standard that supports QinQ networks by allowing multiple 802.1Q tags in an Ethernet frame.

  Refer to Chapter 11, starting on page 385.

1.1.3 Layer 3 Software Features

Arista switches support these layer 3 software features:

- **Internet Protocols:** IPv4 and IPv6 are Internet layer protocols that define packet-switched internetworking.

  — IPv4 is described in Chapter 17, starting on page 681.
  — IPv6 is described in Chapter 18, starting on page 727.

- **Equal Cost Multi-Path Routing (ECMP)**: ECMP Routing balances traffic over multiple paths. Refer to Section 10.4.2, Equal Cost Multipath Routing (ECMP) and Load Sharing, on page 330.

- **Open Shortest Path First Protocol, version 2 (OSPFv2)**: OSPFv2 is a link-state routing protocol used by IPv4 networks to route packets within a single routing domain. Refer to Chapter 19, starting on page 763.

- **Open Shortest Path First Protocol, version 3 (OSPFv3)**: OSPFv3 is a link-state routing protocol used by IPv6 networks to route packets within a single routing domain. Refer to Chapter 20, starting on page 837.

- **Border Gateway Protocol (BGP)**: BGP is an Internet routing protocol that maintains network accessibility among autonomous systems. Refer to Chapter 21, starting on page 889.

- **Routing Information Protocol (RIP)**: RIP is a distance vector routing protocol typically used as an interior gateway protocol. Refer to Chapter 22, starting on page 975.

- **Multicast Services**: Multicast Services support the simultaneous delivery of information to a group of destinations where messages are delivered over each link of the network only once and data is copied only when links to multiple destinations split. Refer to Chapter 23, starting on page 991.

- **Static Routing**: Arista switches support fixed network address assignments to routers and other network devices. Refer to Section 10.2.2.1, Static Routing, on page 325.

- **VPN Routing and Forwarding (VPN):** VPN supports data traffic separation by maintaining multiple routing tables. Refer to Section 10.4.3, Routing Tables / VPN Routing and Forwarding (VRF), on page 330.

1.2 Feature Availability on Switch Platforms

The tables in this section list the features that are supported by each Arista switch platform.

1.2.1 Management Features

Feature	7100 Series	7500 Series	7048	7050 Series
Industry Standard CLI	YES	YES	YES	YES
In-band management	YES	YES	YES	YES
SSH v2	YES	YES	YES	YES
Telnet	YES	YES	YES	YES
Control-Plane Access Control Lists (CP-ACL)	YES	YES	YES	YES
TACACS+ Authentication and Authorization (PAP)	YES	YES	YES	YES
TACACS+ Accounting	YES	YES	YES	YES
RADIUS Authentication	YES	YES	YES	YES
RADIUS Accounting	YES	YES	YES	YES
Management port isolation	YES	YES	YES	YES
DNS Client	YES	YES	YES	YES
NTP	YES	YES	YES	YES
IEEE 802.1AB LLDP	YES	YES	YES	YES
Syslog	YES	YES	YES	YES
File download via SCP, HTTP, HTTPS, FTP, and TFTP	YES	YES	YES	YES
Login and MOTD banners	YES	YES	YES	YES
Interface range support	YES	YES	YES	YES
Show reload cause	YES	YES	YES	YES
Management to IPv6 addresses on VLAN and Management interfaces	YES	YES	YES	YES
VM on EOS	YES	YES	YES	YES
VMTracer	YES	YES	YES	YES
Locator LED	YES	YES	YES	YES
Digital Optical Monitoring (DOM)	YES	YES	YES	YES
Zero Touch Provisioning (ZTP)	YES	YES	YES	YES
ACL counters and logging	YES	NO	NO	YES
CLI Scheduler	YES	YES	YES	YES
Event Manager	YES	YES	YES	YES
Event Monitor	YES	YES	YES	YES
Tcpdump sessions	YES	YES	YES	YES
LANZ	YES	YES	YES	NO
User-Configurable Hardware Control Plane Policing	NO	NO	NO	YES
Management VRF	YES	YES	YES	YES

Table 1-1 Management Feature Support

1.2.2 Layer 2 Features

Feature	7100 Series	7500 Series	7048	7050 Series
VLAN based port segmentation	YES	YES	YES	YES
Tagged native VLAN mode	YES	YES	YES	YES
IEEE 802.1D Bridging	YES	YES	YES	YES
IEEE 802.1Q Trunking	YES	YES	YES	YES
IEEE 802.1ad QinQ	YES	NO	NO	YES
IEEE 802.1w RSTP (Rapid Spanning Tree Protocol)	YES	YES	YES	YES
IEEE 802.1s MSTP (Multiple Spanning Tree Protocol)	YES	YES	YES	YES
Rapid Per VLAN Spanning Tree Protocol	YES	YES	YES	YES
BPDU Guard	YES	YES	YES	YES
BPDU filtering	YES	YES	YES	YES
Disable STP on a VLAN to support Routed Ports	YES	YES	YES	YES
Backup Interface	YES	YES	YES	YES
Link Aggregation Groups (up to 16 ports)	YES	YES	YES	YES
Link Aggregation hash utilizing L2 & L3 packet header fields	YES	YES	YES	YES
IEEE 802.3ad LACP (Link Aggregation Control Protocol)	YES	YES	YES	YES
Multi-chassis Link Aggregation (MLAG)	YES	YES	YES	YES
IGMP Snooping + MLAG	YES	YES	NO	YES
VARP for MLAG	YES	YES	YES	YES
Port mirroring	YES	YES	YES	YES
Port-channel source for port mirroring	YES	YES	YES	YES
MAC security	YES	YES	YES	YES
Layer 2 Access Lists	YES	YES	YES	YES
IEEE 802.1Qaz DCBX (Data Center Bridge Exchange)	YES	NO	NO	YES
IEEE 802.1Qbb PFC (Priority-based Flow Control)	YES	NO	NO	YES
Interface rate counters	YES	YES	YES	YES
mac-address-table configuration	YES	YES	YES	YES
Auto-negotiation with 1000BASE-X	YES	YES	YES	YES
IEEE 802.3x PAUSE frames	YES	YES	YES	YES
Jumbo frames up to 9216 bytes	YES	YES	YES	YES
Sflow	YES	YES	YES	YES
Storm control	YES	NO	NO	YES
Root guard	YES	YES	YES	YES
Loop guard	YES	YES	YES	YES
Bridge assurance	YES	YES	YES	YES
Static MAC multicast	YES	NO	NO	YES
QoS interface trust	YES	YES	YES	YES

Table 1-2 Layer 2 Feature Support

Feature	7100 Series	7500 Series	7048	7050 Series
Egress port shaping	YES	YES	YES	YES
Egress queue scheduling and shaping	YES	YES	YES	YES
Private VLANs	YES	NO	NO	YES
DSCP Rewrite	YES	NO	NO	YES

Table 1-2 Layer 2 Feature Support

1.2.3 Layer 3 Features

Feature	7100 Series	7500 Series	7048	7050 Series
IPv4 Static Routing	YES	YES	YES	YES
IPv6 Static Routing	NO	YES	YES	YES
Routed Interfaces	YES	YES	YES	YES
L3 Multipathing / Equal Cost Multi-Path routing (ECMP)	YES	YES	YES	YES
Interfaces per ECMP group	16	16	16	32
OSPFv2	YES	YES	YES	YES
OSPFv3	NO	YES	YES	YES
BGPv4	YES	YES	YES	YES
MP-BGP IPv6	NO	YES	YES	YES
Layer 3 Access Control Lists	YES	YES	YES	YES
Router Access Control Lists	NO	NO	NO	YES
DHCP Relay	YES	YES	YES	YES
Static ARP entries	YES	YES	YES	YES
Route Maps	YES	YES	YES	YES
RIPv2	YES	YES	YES	YES
Loopback interfaces	YES	YES	YES	YES
NULL interface	YES	YES	YES	YES

Table 1-3 Layer 3 Feature Support

Chapter 2

Initial Configuration and Recovery

This chapter describes initial configuration and recovery tasks. Later chapters provide details about features introduced in this chapter.

This chapter contains these sections:

- Section 2.1: Initial Switch Access
- Section 2.2: Connection Management
- Section 2.3: Recovery Procedures
- Section 2.4: Upgrades
- Section 2.5: Session Management Commands

2.1 Initial Switch Access

Arista Network switches provide two initial configuration methods:

- Zero Touch Provisioning configures the switch without user interaction (Section 2.1.1).
- Manual provisioning configures the switch through commands entered by a user through the CLI (Section 2.1.2).

2.1.1 Zero Touch Provisioning

Zero Touch Provisioning (ZTP) configures a switch without user intervention by downloading a startup configuration file (***startup-config***) or a boot script from a location specified by a DHCP server. Section 6.3.4 describes network tasks required to set up ZTP.

The switch enters ZTP mode when it boots if flash memory does not contain ***startup-config***. It remains in ZTP mode until a user cancels ZTP mode or until the switch retrieves a ***startup-config*** or a boot script. After downloading a file through ZTP, the switch reboots again, using the retrieved file.

To provision the switch through Zero Touch Provisioning:

Step 1 Mount the switch in its permanent location.

Step 2 Connect at least one management or Ethernet port to a network that can access the DHCP server and configuration file.

Step 3 Provide power to the switch.

ZTP provisioning progress can be monitored through the console port. Section 2.1.2.1 provides information for setting up the console port. Section 2.1.2.2 provides information for monitoring ZTP progress and cancelling ZTP mode.

2.1.2 Manual Provisioning

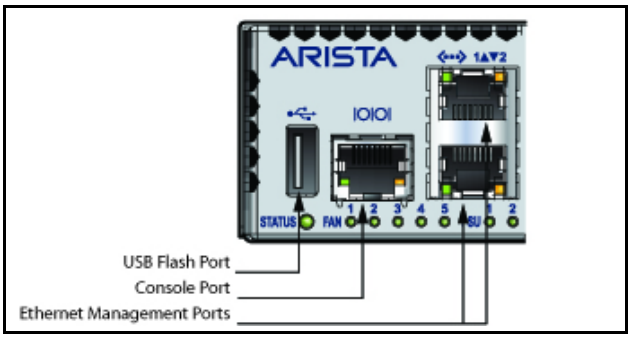
Initial manual switch provisioning requires the cancellation of ZTP mode, the assignment of an IP address to a network port, and the establishment of an IP route to a gateway. Initial provision is performed through the serial console and Ethernet management ports.

- The console port provides serial access to the switch. These conditions may require serial access:
  - management ports are not assigned IP addresses
  - the network is inoperable
  - the enable password is not available
- The Ethernet management ports are used for out of band network management tasks. Before using a management port for the first time, an IP address must be assigned to that port.

2.1.2.1 Console Port

The console port is a serial port located on the front of the switch. Figure 2-1 shows the console port on the 7124-S switch. You can connect a PC or terminal to the console port through a serial or RS-232 cable. The accessory kit includes an RJ-45 to DB-9 adapter cable for connecting the switch.

Figure 2-1 Switch Ports



Port Settings

When connecting a PC or terminal to the console port, use these settings:

- 9600 baud
- no flow control
- 1 stop bit
- no parity bits
- 8 data bits

Admin Username

The initial configuration provides one username, **admin,** that is not assigned a password. When using the admin username without a password, you can only log into the switch through the console port. After a password is assigned to the **admin** username, it can log into the switch through any port.

The **username** command assigns a password to the specified username.

Example

- This command assigns the password **pxq123** to the **admin** username:

```
Switch(config)#username admin secret pxq123
Switch(config)#
```

The **admin** username is now password protected and can log into the switch from any port.

New and altered passwords that are not saved to the startup configuration file, as described in Section 3.5.3: Saving the Running Configuration Settings, are lost when the switch is rebooted.

2.1.2.2 Cancelling Zero Touch Provisioning

Zero Touch Provisioning (ZTP) installs a ***startup-config*** file from a network location if flash memory does not contain a ***startup-config*** when the switch reboots. Cancelling ZTP is required if the switch cannot download a ***startup-config*** or boot script file.

When the switch boots without a ***startup-config*** file, it displays the following message through the console port:

```
No startup-config was found.

The device is in Zero Touch Provisioning mode and is attempting to
download the startup-config from a remote system. The device will not
be fully functional until either a valid startup-config is downloaded
from a remote system or Zero Touch Provisioning is cancelled. To cancel
Zero Touch Provisioning, login as admin and type 'zerotouch cancel'
at the CLI.

localhost login:
```

To cancel ZTP mode, log into the switch with the **admin** password, then enter the **zerotouch cancel** command. The switch immediately boots without installing a ***startup-config*** file.

```
localhost login: admin
admin
localhost>Apr 15 21:28:21 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP
request on  [ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18,
Ethernet21, E-thernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9,
Management1, Management2 ]
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-DHCP_QUERY_FAIL: Failed to get a valid
DHCP response
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-RETRY: Retrying Zero Touch
Provisioning from the beginning (attempt 1)
Apr 15 21:29:22 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP request on
[ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18, Ethernet21,
Ethernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9, Management1,
Management2 ]


localhost>zerotouch cancel
zerotouch cancel
localhost>Apr 15 21:29:39 localhost ZeroTouch: %ZTP-5-CANCEL: Cancelling Zero
Touch Provisioning
Apr 15 21:29:39 localhost ZeroTouch: %ZTP-5-RELOAD: Rebooting the system
Broadcast messagStopping sshd: [  OK  ]
watchdog is not running
SysRq : Remount R/O
Restarting system
ø

Aboot 1.9.0-52504.EOS2.0
Press Control-C now to enter Aboot shell
```

Section 6.3.1 lists the remaining messages that the switch displays before providing a logon prompt. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file as described by step 8 of Section 2.1.2.3.

2.1.2.3 Ethernet Management Port

Arista switches provide one or two Ethernet management ports for configuring the switch and managing the network out of band. Figure 2-1 shows the location of the Ethernet management ports. Only one port is required to manage the switch – when available, the second port provides redundancy.

You can access the Ethernet management ports remotely over a common network or locally through a directly connected PC. Before you can access the switch through a remote connection, an IP address and a static route to the default gateway is required.

Assigning an IP Address to an Ethernet Management Port

This procedure assigns an IP address to an Ethernet management port:

Step 1 Connect a PC or terminal server to the console port.

Use the settings listed in Section 2.1.2.1 under Port Settings.

Step 2 Type **admin** at the login prompt to log into the switch. The initial login does not require a password.

```
Arista EOS
Switch login:admin
Last login: Fri Apr 9 14:22:18 on Console

Switch>
```

Step 3 Type **enable** at the command prompt to enter Privileged EXEC mode. See Section 3.4.1: Mode Types for information about Privileged EXEC mode.

```
Switch>enable
Switch#
```

Step 4 Type **configure terminal** (or **config**) to enter global configuration mode. See Section 3.4.1: Mode Types for information about global configuration mode.

```
Switch#configure terminal
Switch(config)#
```

Step 5 Type **interface management 1** to enter Interface Configuration mode.

Any available management port can be used in place of management port 1.

```
Switch(config)#interface management 1
Switch(config-if-Ma1)#
```

Step 6 Type **ip address**, followed by the desired address, to assign an IP address to the port.

This command assigns the IP address 192.0.2.8 to management 1 port.

```
Switch(config-if-Ma1)#ip address 192.0.2.8/24
```

Step 7 Type **end** at the Interface Configuration and global configuration prompts to return to Privileged EXEC mode.

```
Switch(config-if-Ma1)#end
Switch(config)#end
Switch#
```

Step 8 Type **write memory** (or **copy running-config startup-config)** to save the new configuration to the ***startup-config*** file. See Section 3.5.3: Saving the Running Configuration Settings.

```
Switch# write memory
Switch#
```

Configuring a Default Route to the Gateway

This procedure configures a default route to a gateway located at 192.0.2.1.

Step 1 Enter global configuration mode.

```
Switch>enable
Switch#configure terminal
Switch(config)#
```

Step 2 Create a static route to the gateway with the IP route command.

```
Switch(config)#ip route 0.0.0.0/0 192.0.2.1
```

Step 3 Save the new configuration.

```
Switch#write memory
Switch#
```

2.2 Connection Management

The switch supports three connection methods:

- console
- SSH
- Telnet

The switch always enables console and SSH. Telnet is disabled by default.

The **management** command places the switch in a configuration mode for changing the idle timeout period. The idle timeout period determines the inactivity interval that terminates a connection session. Telnet sessions are enabled from management telnet configuration mode.

Examples

- The **management console** command places the switch in console management mode:

```
switch(config)#management console
switch(config-mgmt-console)#
```

- The **management ssh** command places the switch in SSH management mode:

```
switch(config)#management ssh
switch(config-mgmt-ssh)#
```

- The **management telnet** command places the switch in Telnet management mode:

```
switch(config)#management telnet
switch(config-mgmt-telnet)#
```

- The **exit** command returns the switch to global configuration mode.

```
switch(config-mgmt-ssh)#exit
switch(config)#
```

The **idle-timeout** command configures the idle-timeout period for the connection method designated by the active configuration mode. The default idle timeout period for each connection method is 60 minutes.

Examples

- This command configures an ssh idle-timeout period of three hours.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#idle-timeout 180
```

- This command returns the console idle-timeout period to the default 60 minute setting.

```
switch(config)#management console
switch(config-mgmt-console)#idle-timeout 60
```

The **shutdown (Management-Telnet)** command enables and disables Telnet connections.

Examples

- These commands enable Telnet.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#no shutdown
```

- These commands disable Telnet.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#shutdown
```

2.3 Recovery Procedures

These sections describe switch recovery procedures:

- Section 2.3.1: Removing the Enable Password from the Startup Configuration
- Section 2.3.2: Reverting the Switch to the Factory Default Startup Configuration
- Section 2.3.3: Restoring the Factory Default EOS Image and Startup Configuration
- Section 2.3.4: Restoring the Configuration and Image from a USB Flash Drive

The first three procedures require Aboot Shell access through the console port. If the console port is not accessible, use the last procedure in the list to replace the configuration file through the USB Flash Drive.

Chapter 6, starting on page 197 describes the switch booting process and includes descriptions of the Aboot shell, Aboot boot loader, and required configuration files.

2.3.1 Removing the Enable Password from the Startup Configuration

The **enable password** controls access to Privileged EXEC mode. To prevent unauthorized disclosure, the switch stores the **enable password** as an encrypted string that it generates from the clear text password. When the switch authentication mode is local and an **enable password** is configured, the CLI prompts the user to enter the clear text password after the user types **enable** at the EXEC prompt.

The ***startup-config*** file stores the encrypted **enable password** to ensure that the switch loads it when rebooting. If the text version of the **enable password** is lost or forgotten, access to enable mode is restored by removing the encrypted **enable password** from the startup configuration file.

This procedure restores access to enable mode without changing any other configuration settings.

Step 1 Access the Aboot shell:

Step a Power cycle the switch by successively removing and restoring access to its power source.

Step b Type **Ctrl-C** when prompted, early in the boot process.

Step c Enter the Aboot password, if prompted.

If the Aboot password is unknown, refer to Section 2.3.3: Restoring the Factory Default EOS Image and Startup Configuration for instructions on reverting all flash directory contents to the factory default, including the startup configuration and EOS image.

Step 2 Change the active directory to /mnt/flash directory.

```
Aboot#cd /mnt/flash
```

Step 3 Open the startup-config file in vi.

```
Aboot#vi startup-config
```

Step 4 Remove the enable password line.

This is an example of an enable password line:

```
enable secret 5 $1$dBXo2KpF$Pd4XYLpI0ap1ZaU7glG1w/
```

Step 5 Save the changes and exit vi.

Step 6 Exit Aboot. This boots the switch.

```
Aboot#exit
```

Refer to Section 4.2.1.4: Enable Command Authentication for information on the **enable password**.

2.3.2 Reverting the Switch to the Factory Default Startup Configuration

The ***startup-config*** file contains configuration parameters that the switch uses during a boot. Parameters that do not appear in ***startup-config*** are set to their factory defaults when the switch reloads. The process requires the Aboot password if Aboot is password protected.

This procedure reverts EOS configuration settings to the default state through bypassing the ***startup-config*** file during a switch boot.

Step 1 Access the Aboot shell through the console port:

Step a Type **reload** at the Privileged EXEC prompt.

Step b Type **Ctrl-C** when prompted, early in the boot process.

Step c Enter the Aboot password, if prompted.

If the Aboot password is unknown, refer to Section 2.3.3: Restoring the Factory Default EOS Image and Startup Configuration for instructions on reverting all flash directory contents to the factory default, including ***startup-config*** and EOS image.

Step 2 Change the active directory to **/mnt/flash** directory.

```
Aboot#cd /mnt/flash
```

Step 3 Rename the startup configuration file.

```
Aboot#mv startup-config startup-config.old
```

Step 4 Exit Aboot. This boots the switch

```
Aboot#exit
```

Step 5 Cancel Zero Touch Provisioning (ZTP). Refer to Section 2.1.2.2: Cancelling Zero Touch Provisioning for instructions.

If ZTP is not cancelled, the switch either:

- boots, using the ***startup-config*** file or boot script that it obtains from the network, or
- remains in ZTP mode if the switch is unable to download a ***startup-config*** file or boot script.

Step 6 Configure the **admin** and **enable** passwords.

Refer to Section 4.2.1: Local for information about creating usernames and passwords.

```
Switch>enable
Switch#configure terminal
Switch(config)#enable secret xyz1
Switch(config)#username admin secret abc41
```

Step 7 Save the new ***running-config*** to the startup configuration file.

```
Switch#write memory
```

Step 8 (Optional) Delete the old startup configuration file.

```
Switch#delete startup-config.old
```

After ZTP is cancelled, the switch reboots, using the factory default settings. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file before the next switch reboot.

2.3.3 Restoring the Factory Default EOS Image and Startup Configuration

A **fullrecover** command removes all internal flash contents (including configuration files, EOS image files, and user files), then restores the factory default EOS image and ***startup-config***. A subsequent installation of the current EOS image may be required if the default image is outdated. This process requires Aboot shell access through the console port.

This procedure restores the factory default EOS image and startup configuration.

Step 1 Access the Aboot shell through the console port:

Step a Type **reload** at the Privileged EXEC prompt.

Step b Type **Ctrl-C** when prompted, early in the boot process.

Step c Enter the Aboot password, if prompted.

If the Aboot password is not known, enter an empty password three times, after which the CLI displays:

```
Type "fullrecover" and press Enter to revert /mnt/flash to factory default
state, or just press Enter to reboot:
```

Type **fullrecover** and go to step 4.

Step 2 Type **fullrecover** at the Aboot prompt.

```
Aboot#fullrecover
```

Aboot displays this warning:

```
All data on /mnt/flash will be erased; type "yes" and press Enter to
proceed, or just press Enter to cancel:
```

Step 3 Type **yes** and press **Enter**.

The switch performs these actions:

- erases the contents of /mnt/flash
- writes new boot-config, startup-config, and EOS.swi files to /mnt/flash
- returns to the Aboot prompt

Step 4 Exit Aboot. This boots the switch.

```
Aboot#exit
```

The serial console settings are restored to their default values (9600/N/8/1/N).

Step 5 Reconfigure the console port if non-default settings are required.

Step 6 Cancel Zero Touch Provisioning (ZTP). Refer to Section 2.1.2.2: Cancelling Zero Touch Provisioning for instructions.

If ZTP is not cancelled, the switch either:

- boots, using the ***startup-config*** file or boot script that it obtains from the network, or
- remains in ZTP mode if the switch is unable to download a ***startup-config*** file or boot script.

After ZTP is cancelled, the switch reboots, using the factory default settings. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file before the next switch reboot.

2.3.4 Restoring the Configuration and Image from a USB Flash Drive

The USB flash drive port can be used to restore an original configuration when you cannot establish a connection to the console port. This process removes the contents of the internal flash drive, restores the factory default configuration, and installs a new EOS image from the USB flash drive.

This procedure restores the factory default configuration and installs an EOS image stored on a USB flash drive.

Step 1 Prepare the USB flash drive:

Step a Verify the drive is formatted with MS-DOS or FAT file system.

Most USB drives are pre-formatted with a compatible file system.

Step b Create a text file named **fullrecover** on the USB flash drive.

The filename does not have an extension. The file may be empty.

Step c Create a text file named **boot-config**.

The last modified timestamp of the **boot-config** file on the USB flash must differ from the timestamp of the **boot-config** file on the switch.

Step d Enter this line in the new **boot-config** file on the USB flash:

```
SWI=flash:EOS.swi
```

Step e Copy an EOS image file to the flash drive. Rename it **EOS.swi** if it has a different file name.

For best results, the flash drive should contain only these three files because the procedure copies all files and directories on the USB flash drive to the switch.

- fullrecover
- boot-config
- EOS.swi

Step 2 Insert the USB flash drive into the USB flash port on the switch, as shown in Figure 2-1.

Step 3 Connect a terminal to the console port and configure it with the default terminal settings (9600/N/8/1) to monitor progress messages on the console.

Step 4 Power up or **reload** the switch.

The switch erases internal flash contents and copies the files from the USB flash drive to internal flash. The switch then boots automatically.

Step 5 Cancel Zero Touch Provisioning (ZTP). Refer to Section 2.1.2.2: Cancelling Zero Touch Provisioning for instructions.

If ZTP is not cancelled, the switch either:

- boots, using the ***startup-config*** file or boot script that it obtains from the network, or
- remains in ZTP mode if the switch is unable to download a ***startup-config*** file or boot script.

After ZTP is cancelled, the switch reboots, using the factory default settings. To avoid entering ZTP mode on subsequent reboots, create a ***startup-config*** file before the next switch reboot.

2.4 Upgrades

The active EOS image on a switch is updated by the **boot system** command. This command can load an image file from one of various locations to update or downgrade the switch to any available image.

Modifying the active EOS image is a four step process:

1. Transfer the image file to the switch (Section 2.4.1).

   This step is not necessary if the desired image file is on the switch.
2. Modify the boot-config file to point at the desired image file (Section 2.4.2).
3. Reload the switch (Section 2.4.3).
4. Verify the switch is running the new image (Section 2.4.4).

2.4.1 Transferring the Image File

The desired image must be loaded to the file system on the switch, typically into the flash. Use the CLI **copy** command to load files to the flash.

These command examples transfer an image file to flash from various locations.

USB Memory

Command

```
copy usb1:/sourcefile flash:/destfile
```

Example

```
Sch#copy usb1:/EOS-4.9.2.swi flash:/EOS-4.9.2.swi
```

FTP Server

Command

```
copy ftp:/ftp-source/sourcefile flash:/destfile
```

Example

```
Sch#copy ftp:/user:password@10.0.0.3/EOS-4.9.2.swi flash:/EOS-4.9.2.swi
```

SCP

Command

```
copy scp://scp-source/sourcefile flash:/destfile
```

Example

```
Sch#copy scp://user:password@10.1.1.8/user/EOS-4.9.2.swi flash:/EOS-4.9.2.swi
```

HTTP

Command

```
copy http://http-source/sourcefile flash:/destfile
```

Example

```
Sch#copy http://10.0.0.10/EOS-4.9.2.swi flash:/EOS-4.9.2.swi
```

2.4.2 Modify boot-config

When the switch boots, the Aboot process reads the **boot-config** file to select an image file. After transferring the desired image file, use the **boot system** command to update the boot-config file.

This command changes the boot-config file to point at the image file located in flash memory at EOS-4.9.2.swi.

```
Switch#configure terminal
Switch(config)#boot system flash:/EOS-4.9.2.swi
```

Use the **show boot-config** command to verify that the boot-config file is correct:

```
Switch(config)#show boot-config
Software image: flash:/EOS-4.9.2.swi
Console speed: (not set)
Aboot password (encrypted): $1$ap1QMbmz$DTqsFYeauuMSa7/Qxbi2l1
```

If you modified any running configuration settings, save the configuration to the startup-config file with the **write memory** command.

```
Switch#write memory
```

2.4.3 Reload

After updating the boot-config file, reload the switch to activate the new image. The **reload** command reloads the switch.

The EOS displays this text from any port except the console. When reloading from the console port, all rebooting messages are displayed on the terminal. See Section 6.3: System Reset for information about rebooting the system.

```
Switch#reload
The system is going down for reboot NOW!
```

2.4.4 Verify

After the switch finishes reloading, log into the switch and use the **show version** command to confirm the correct image is loaded. The ***Software image version*** line displays the version of the active image file.

```
Switch#show version
Arista DCS-7124S
Hardware version: 03.04
Serial number: JFL07340036

Software image version: 4.9.2
Architecture: i386
Internal build version: 4.9.2-59039.EOS4.9.2
Internal build ID: f34b0734-30ea-4544-b8c2-679b1b6beccf

Uptime: 1 minute
Total memory: 1015232 kB
Free memory: 14440 kB
```

2.5 Session Management Commands

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Commands

- management . . . . . Page 49

Management Configuration Commands

- idle-timeout . . . . . Page 48
- shutdown (Management-Telnet) . . . . . Page 51

Inventory Display Command

- show inventory . . . . . Page 50

idle-timeout

The **idle-timeout** command configures the connection timeout period for the connection type denoted by the active connection management mode. The connection timeout period defines the interval between a user's most recently entered command and an automatic connection shutdown.

The default idle-timeout period is 60 minutes.

Command Modes

Management console configuration
Management ssh configuration
Management telnet configuration

Command Syntax

idle-timeout *idle_period*

Parameters

- *idle_period* session idle timeout length (minutes). Values range from 0 to 86400 (24 hours).

Example

- These commands configure an ssh idle-timeout period of three hours, then returns the switch to global configuration mode.

```
switch(config)#management ssh
switch(config-mgmt-ssh)#idle-timeout 180
switch(config-mgmt-ssh)#exit
switch(config)#
```

- These commands returns the console idle-timeout period to the default 60 minute setting.

```
switch(config)#management ssh
switch(config-mgmt-console)#idle-timeout 60
switch(config-mgmt-console)#
```

management

The **management** command places the switch in a management configuration mode to adjust the idle timeout period or to enable Telnet. The idle timeout period determines the inactivity interval that terminates a connection session. The default idle timeout period is 60 minutes.

The switch provides three management configuration modes:

- console management
- ssh management
- Telnet management

Commands available in the management modes include

- **exit**
- **idle-timeout**
- **shutdown (Management-Telnet)** (Telnet management mode only)

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

management *session_type*
exit

Parameters

- *session_type* communication session method. Options include:
  - **console**
  - **ssh**
  - **telnet**

Example

- This command places the switch in console management mode:

```
switch(config)#management console
switch(config-mgmt-console)#
```

- This command places the switch in ssh management mode:

```
switch(config)#management ssh
switch(config-mgmt-ssh)#
```

- This command places the switch in Telnet management mode:

```
switch(config)#management telnet
switch(config-mgmt-telnet)#
```

- This command returns the switch to global management mode:

```
switch(config-mgmt-telnet)#exit
switch(config)#
```

show inventory

The **show inventory** command displays the hardware components installed in the switch. Serial numbers and a description is also provided for each component.

Command Mode
EXEC Configuration

Command Syntax
show inventory

Examples

- This command displays the hardware installed in a DCS-7148SX switch.

```
switch>show inventory
System information
  Model          HW Version  Serial Number  Description              Mfg Date
  -------------- ----------- -------------- ------------------------ ----------
  DCS-7148SX     04.05       JFL08130099    48-port SFP+ 10GigE 1RU  2008-04-25

System has 2 power supply slots
  Slot Model            Serial Number
  ---- ---------------- ----------------
  1    PWR-760AC        I080FA005D1YZ
  2    PWR-760AC        I080FH004V1YZ

System has 5 fan modules
  Module  Number of Fans  Model            Serial Number
  ------- --------------- ---------------- ----------------
  1       1               FAN-7100-F       JFL0000000
  2       1               FAN-7100-F       JFL0000000
  3       1               FAN-7100-F       JFL0000000
  4       1               FAN-7100-F       JFL0000000
  5       1               FAN-7100-F       JFL0000000

System has 50 ports
  Type             Count
  ---------------- ----
  Management       2
  Switched         48

System has 48 transceiver slots
  Port Manufacturer     Model            Serial Number    Rev
  ---- ---------------- ---------------- ---------------- ----
  1    Arista Networks  SFP-10G-SRL      XCW1053FE12R     0002
  2    Arista Networks  SFP-10G-SRL      XCW1044FE1D2     0002
<-------OUTPUT OMITTED FROM EXAMPLE-------->
  47   Arista Networks  SFP-10G-SRL      XCW1039FE0D8     0002
  48   Arista Networks  SFP-10G-SRL      XCW1103FE02E     0002

switch>
```

shutdown (Management-Telnet)

The **shutdown** command, in management-telnet mode, disables or enables Telnet on the switch. Telnet is disabled by default. The **management** command places the switch in management-telnet mode.

- To enable Telnet, enter **no shutdown** at the management-telnet prompt.
- To disable Telnet, enter **shutdown** at the management-telnet prompt.

Command Modes

Management-Telnet Configuration

Command Syntax

```
shutdown
no shutdown
```

Example

- These commands enable Telnet, then returns the switch to global configuration mode.

```
switch(config)#management telnet
switch(config-mgmt-telnet)#no shutdown
switch(config-mgmt-telnet)#exit
switch(config)#
```

- This command disables Telnet.

```
switch(config-mgmt-telnet)#shutdown
```

Chapter 3

Command-Line Interface

The Extensible Operating System (EOS) provides the interface for entering commands that control the switch and manage the network. This chapter describes the command-line interfaces (CLI) that access the switch.

This chapter includes these sections:

- Section 3.1: Accessing the EOS CLI
- Section 3.2: Processing Commands
- Section 3.3: Switch Platforms
- Section 3.4: Command Modes
- Section 3.5: Managing Switch Configuration Settings
- Section 3.6: Other Command-Line Interfaces
- Section 3.7: Directory Structure
- Section 3.8: Command-Line Interface Commands

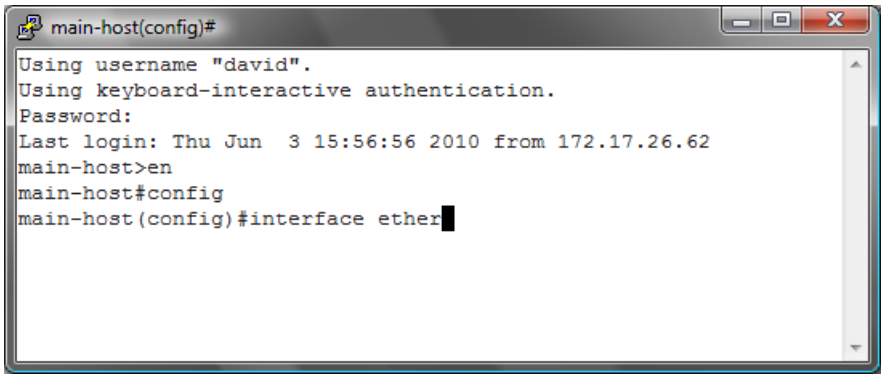
3.1 Accessing the EOS CLI

You can open an EOS CLI session through these connections:

- Ethernet Management Ports
- Console Port
- Telnet Connections
- Secure Shell (SSH)

Figure 3-1 displays the EOS CLI in a Secure Shell connection.

Figure 3-1 EOS Command-Line Interface



3.2 Processing Commands

3.2.1 Command Execution

Command keywords are not case sensitive. The CLI accepts truncated keywords that uniquely correspond to one command.

- The command abbreviation **con** does not execute a command in Privileged EXEC mode because the names of two commands begin with these letters: **configure** and **connect**.

```
Switch#con
% Ambiguous command
```

- The command abbreviation **conf** executes **configure** in Privileged EXEC mode because no other command name begins with **conf**.

```
Switch#conf
Switch(config)#
```

3.2.2 Alias

The **alias** command creates an alias for a CLI command. Entering the alias in the CLI executes the corresponding command.

Example

- This command makes **srie** an alias for the command **show running-config interface ethernet 1-5**

```
Switch(config)#alias srie show running-config interface ethernet 1-5
Switch(config)#srie
interface Ethernet1
   switchport access vlan 33
   storm-control broadcast level 1
   spanning-tree portfast
   spanning-tree bpduguard enable
interface Ethernet2
   switchport access vlan 33
   spanning-tree portfast
interface Ethernet3
   switchport access vlan 33
   spanning-tree portfast
   spanning-tree bpduguard enable
interface Ethernet4
interface Ethernet5
   shutdown
```

3.2.3 Cursor Movement Keystrokes

EOS supports these cursor movement keystrokes:

- **Ctrl-B** or the **Left Arrow** key: Moves the cursor back one character.
- **Ctrl-F** or the **Right Arrow** key: Moves the cursor forward one character.
- **Ctrl-A**: Moves the cursor to the beginning of the command line.
- **Ctrl-E**: Moves the cursor to the end of the command line.
- **Esc-B**: Moves the cursor back one word.
- **Esc-F**: Moves the cursor forward one word.

3.2.4 History Substitution Keystrokes

The history buffer retains the last 20 entered commands. History substitution keystrokes that access previously entered commands include:

- **Ctrl-P** or the **Up Arrow** key: Recalls history buffer commands, beginning with the most recent command. Repeat the key sequence to recall older commands.
- **Ctrl-N** or the **Down Arrow** key: Returns to more recent commands after using the **Ctrl-P** or the **Up Arrow**. Repeat the key sequence to recall more recent commands.

The **show history** command in Privileged EXEC mode displays the history buffer contents.

```
SwitchName#show history
  en
  config
  exit
  show history
```

3.2.5 Command Lists and Syntax Assistance

EOS CLI uses widely followed conventions for providing command lists and syntax assistance. These conventions are available in all command modes.

- To display a list of available commands, type a question mark (**?**):

```
SwitchName>?
  clear       Reset functions
  connect     Open a terminal connection
  disable     Turn off privileged commands
  enable      Turn on privileged commands
  exit        Exit from the EXEC
  help        Description of the interactive help system
  logout      Exit from the EXEC
  no          Negate a command or set its defaults
  ping        Send echo messages
  show        Show running system information
  telnet      Open a telnet connection
  terminal    Configure the terminal
  traceroute  Trace route to destination
```

- To display a list of commands beginning with a specific character sequence, type the sequence followed by a question mark.

```
Switch#di?
diagnostic  diff  dir  disable
```

- To display a command's keywords or arguments, type a question mark as an argument.

```
Switch>ping ?
  WORD  Ping destination address or hostname
```

- The switch accepts an address-mask or CIDR notation (address-prefix) in commands that require an IP address and mask. These commands are processed identically:

```
switch(config)#ip route 0.0.0.0 255.255.255.255 10.1.1.254

switch(config)#ip route 0.0.0.0/32 10.1.1.254
```

- The switch accepts an address-wildcard or CIDR notation in commands requiring an IP address and wildcard. Wildcards use zeros to mask portions of the IP address and is found in some protocol configuration statements, including OSPF. The switch processes these commands identically:

```
switch:network 10.255.255.1 0.0.0.255 area 15

switch:network 10.255.255.1/24 area 15
```

3.2.6 Regular Expressions

A regular expression is pattern of symbols, letters, and numbers that represent an input string for matching an input string entered as a CLI parameter. The switch uses regular expression pattern matching in several BGP commands.

Regular expressions use the following operands:

. (period) matches any single character.

Example **1.3** matches 123, 133, and 1c3.

\ **(backslash)** matches character or special character following the backslash.

Example **15\.5\..** matches 15.5.10.10 it does not match 15.52.10.10
Example **\.** matches . (period)

^ **(caret)** matches the character or null string at the beginning of a string.

Example **^read** matches reader **^read** does not match bread.

*** (asterisk)** matches zero or more sequences of character preceding the asterisk.

Example 12* matches 167, 1267, or 12267 it does not match 267

+ (plus sign) matches one or more sequences of character preceding the plus sign.

Example **46+** matches 2467 or 24667 it does not match 247

$ (dollar sign) dollar sign matches the character or null string at the end of an input string.

Example **read$** matches bread **read$** but not reads

[ ] (brackets) matches characters or a character range separated by a hyphen.

Example **[0137abcr-y]** matches 0, 1, 3,v it does not match 2, 9, m, z

? (question mark) pattern matches zero or one instance. Entering Crtl-V prior to the question mark prevents the CLI from interpreting **?** as a help command.

Example **x1?x** matches ***xx*** and ***x1x***

| (pipe) pattern matches character patterns on either side of bar.

Example **B(E|A)D** matches ***BED*** and ***BAD***. It does not match BD, BEAD, BEED, or EAD

()(parenthesis) nests characters for matching. Endpoints of a range are separated with a dash (-).

Example **6(45)+** matches 645454523 it does not match 6443
Example **([A-Za-z][0-9])+** matches ***C4*** or ***x9***

_ (underscore) Pattern replaces a long regular expression list by matching a comma (,), the beginning of the input string, the end of the input string, or a space.

Example **_rxy_** matches any of the following:

^rxy$
^rxy 23
21 rxy
,rxy,
rxy
,rxy.

The order for matching using the * or + character is longest construct first. Nested constructs are matched from the outside in. Concatenated constructs are matched beginning at the left side. If a regular expression can match two different parts of an input string, it matches the earliest part first.

3.2.7 Scheduling CLI Commands

The **schedule** command facilitates the periodic execution of a specified CLI command. Command parameters configure the interval between consecutive execution instances and the maximum number of files that can be created when the command requires log files. By default, periodic execution of the following ***show tech-support*** command is enabled:

```
schedule tech-support interval 60 max-log-files 100 command show tech-support
```

Examples

- This command schedules the execution of a script file once every 12 hours. The log file option is set to zero because the command does not generate output to std-out.

```
switch#schedule ms_1 interval 720 max-log-files 0 command bash
/mnt/flash/myscript.sh
```

This command displays the commands that are scheduled for periodic execution.

```
switch(config)#show schedule summary
Name             Last   Interval Max log   Log file location
                 time   (mins)   files
---------------- -----  -------- --------  -----------------
tech-support     16:13    60     100       flash:/schedule/tech-support
ms_1             16:28   720      10       flash:/schedule/ms_1
```

- This command stores the ***running-config*** contents to a log file once each hour, creating up to 24 log files.

```
switch#schedule backup-test interval 60 max-log-files 24 command show
running-config
```

3.2.8 Running Bash Shell Commands Automatically with Event Handlers

Event handlers execute a Linux Bash shell command in response to a specific system event. An event handler consists of a Bash command, a trigger and a delay; when the trigger event occurs, the action is scheduled to run after ***delay*** seconds.

To create an event handler, use the **event-handler** command. This creates a new event handler and places the CLI in event handler configuration mode for that handler. Use the **action bash** command to configure a Bash command to run when the handler is triggered, and the **trigger** command to specify the trigger. Event handlers can be triggered either by system booting or by a change in a specified interface's operational status or IP address. To change the delay period between the trigger and the action, use the **delay** command.

When an action is run, certain information is passed to it through environment variables. For the ***boot*** trigger, no variables are set. For the ***interface*** triggers, the following variables are set and passed to the action:

$INTF interface name.

$OPERSTATE current operational status of the specified interface.

$IP-PRIMARY current primary IP address of the specified interface.

To execute more than one Bash command in response to a trigger, create a script containing the desired commands and enter the file path to the script as the argument of the **action bash** command.

To display information about all event handlers or about a specific event handler, use the **show event-handler** command.

To delete an event handler, use the **no** form of the **event-handler** command.

Examples

- These commands create an event handler named "eth_4" which will send email to a specified address when there is a change in the operational status of Ethernet interface 4:

```
switch(config)#event-handler eth_4
switch(config-event-eth_4)#action bash email x@yz.com -s "Et4 $OPERSTATE"
switch(config-event-eth_4)#trigger onintf ethernet 4 operstatus
switch(config-event-eth_4)#delay 60
switch(config-event-eth_4)#exit
switch(config)#
```

  The above handler uses the ***$OPERSTATE*** variable to include the current operational state ("linkup" or "linkdown") in the subject of the email. Note that the action will only function if email has been configured on the switch.

- These commands create an event handler named "onStartup" which will execute a user-defined script 60 seconds after the system boots.

```
switch(config)#event-handler onStartup
switch(config-event-onStartup)#action bash /mnt/flash/startupScript1
switch(config-event-onStartup)#trigger onboot
switch(config-event-onStartup)#delay 60
switch(config-event-onStartup)#exit
switch(config)#
```

  The above handler will also be executed on exiting from event-handler configuration mode.

- This command displays information about all event handlers configured on the system.

```
switch#show event-handler
Event-handler onStartup
Trigger: onBoot delay 60 seconds
Action: /mnt/flash/startupScript1
Last Trigger Activation Time: 1 minutes 51 seconds ago
Total Trigger Activations: 1
Last Action Time: 51 seconds ago
Total Actions: 1

switch#
```

- This command deletes the event handler named "onStartup".

```
switch(config)#no event-handler onStartup
switch(config)#
```

3.3 Switch Platforms

Available features and CLI commands vary by switch platform. CLI options may also vary by switch platform on commands that are available on all platforms. Command descriptions in this manual describe feature availability and command parameters on the basis of switch platform, with the notation of exceptions that exist between different models that use a common platform.

To determine the switch's operating platform, display **show hardware** command parameter options.

Examples

- This command displays the operating platform on a switch operating on the FM4000 platform.

```
Switch#show platform ?
  fm4000  fm4000 chip

Switch#
```

- This command displays the operating platform on a switch operating on the Trident platform.

```
Switch#show platform ?
  trident  Trident chip

Switch#
```

- This command displays the operating platform on a switch operating on the Petra platform.

```
Switch#show platform ?
  fe600   Fe600 fabric chip
  petraA  PetraA switch chip
  sand    Sand platform
Switch#
```

The following is list of supported Arista switches and the operating upon which they operate.

- DCS-7048T Petra
- DCS-7048T-A Petra
- DCS-7050Q-16 Trident
- DCS-7050S-52 Trident
- DCS-7050S-64 Trident
- DCS-7050T-52 Trident
- DCS-7050T-64 Trident
- DCS-7120T-4S FM4000
- DCS-7124FX FM4000
- DCS-7124S FM4000
- DCS-7124SX FM4000
- DCS-7140T-4S FM4000
- DCS-7148S FM4000
- DCS-7148SX FM4000
- DCS-7504 Petra
- DCS-7508 Petra

Section 1.2 lists Arista switch feature availability by switch platform.

3.4 Command Modes

Command modes define the user interface state. Each mode is associated with commands that perform a specific set of network configuration and monitoring tasks.

- Section 3.4.1: Mode Types lists the available modes.
- Section 3.4.2: Navigating Through Command Modes lists mode entry and exit commands.
- Section 3.4.3: Command Mode Hierarchy describes the mode structure.
- Section 3.4.4: Group-Change Configuration Modes describes editing aspects of these modes.

3.4.1 Mode Types

The switch includes these command modes:

- **EXEC**: EXEC mode commands display system information, perform basic tests, connect to remote devices, and change terminal settings. When logging into EOS, you enter EXEC mode.

  EXEC mode prompt: `Switch>`

- **Privileged EXEC**: Privileged EXEC mode commands configure operating and global parameters. The list of Privileged EXEC commands is a superset of the EXEC command set. You can configure EOS to require password access to enter Privileged EXEC from EXEC mode.

  Privileged EXEC mode prompt: `Switch#`

- **Global Configuration**: Global Configuration mode commands configure features that affect the entire system, such as system time or the switch name.

  Global Configuration mode prompt: `Switch(config)#`

- **Interface Configuration**: Interface configuration mode commands configure or enable Ethernet, VLAN, and Port-Channel interface features.

  Interface Configuration mode prompt: `Switch(config-if-Et24)#`

- *Protocol specific mode*: Protocol specific mode commands modify global protocol settings. Protocol specific mode examples include **ACL Configuration** and **Router BGP Configuration**.

  The prompt indicates the active command mode. For example, the Router BGP command prompt is `Switch(config-router-bgp)#`

3.4.2 Navigating Through Command Modes

To change the active command mode, perform one of these actions:

- To enter EXEC mode, log into the switch.
- To enter Privileged EXEC mode from EXEC, type **enable** (or **en**) followed, if prompted, by the enable password:

```
Switch>en
Password:
Switch#
```

- To enter Global Configuration mode from Privileged EXEC, type **configure** (or **config**):

```
Switch#config
Switch(config)#
```

Note EOS supports **copy <url> running-config** in place of the **configure network** command.

- To enter Interface Configuration mode from Global Configuration, type **interface** and the name of the interface to be modified:

```
Switch(config)#interface Et24
Switch(config-if-Et24)#
```

- To enter a protocol specific configuration mode from Global Configuration, type the required command for the desired mode.

```
Switch(config)#router bgp 100
Switch(config-router-bgp)#
```

- To return one level from any configuration mode, type **exit**.

```
Switch(config)#exit
Switch#
```

- To return to Privileged EXEC mode from any configuration mode, type **end** or **Ctrl-Z**.

```
Switch(config-if-Et24))#<Ctrl-z>
Switch#
```

- To return to EXEC mode from Privileged EXEC mode, type **disable** (or **dis**).

```
Switch#dis
Switch>
```

- To exit EOS and log out of the CLI, type **exit** from EXEC mode or Privileged EXEC mode.

```
Switch#exit

login:
```

3.4.3 Command Mode Hierarchy

Command modes are hierarchical. The parent mode of a specified command mode is the mode that contains the command that enters the specified mode.

Example

- EXEC mode contains the **enable** command, which enters Privileged EXEC mode. Therefore, EXEC is the parent mode of Privileged EXEC.

Commands that are executable in a specified command mode include all commands available in the specified mode plus all commands executable from its parent mode.

Example

- EXEC mode includes the **ping** command. EXEC mode is the parent mode of Privileged EXEC mode. Therefore, Privileged EXEC mode includes **ping**.

  Additionally, Privileged EXEC is the parent mode of Global Configuration mode. Therefore, Global Configuration mode also includes **ping**.

Executing a configuration mode command from a child mode may change the active command mode.

Example

- Global Configuration mode contains **interface ethernet** and **ip access-list** commands, which enters Interface Configuration and Access Control List (ACL) Configuration modes, respectively. When Interface Configuration is the active mode, the **ip access-list** command is available and changes the active mode to ACL Configuration.

```
Switch(config)#interface ethernet 1
Switch(config-if-Et1)#ip access-list master-list
Switch(config-acl-master-list)#
```

3.4.4 Group-Change Configuration Modes

Group-change modes apply all changes made during an edit session only after exiting the mode. Changes are stored when the user exits the mode, either through an **exit** or **end** command or through a command that enters a different configuration mode.

The **abort** command discards all changes not previously applied.

Access Control List (ACL) and Multiple Spanning Tree (MST) configuration modes are examples of group-change modes.

3.5 Managing Switch Configuration Settings

3.5.1 Verifying the Running Configuration Settings

running-config is the virtual file that stores the operating configuration. The **show running-config** command displays the ***running-config***. The command is supported in Privileged EXEC mode.

Example

- Type **show running-config** in Privileged EXEC mode. The response in the example is truncated to display only the ip route configured in Section 2.1.2.1.

```
Switch#show running-config
! device: Switch (DCS-7124S, EOS-4.6.0-227198.EOS45)
!
<-------OUTPUT OMITTED FROM EXAMPLE-------->
!
ip route 0.0.0.0/0 192.0.2.1
!
<-------OUTPUT OMITTED FROM EXAMPLE-------->
!
end
Switch#
```

3.5.1.1 Verifying Settings for the Current Mode

To display only the lines of ***running-config*** that affect the current mode, use the **active** option of the **show (various configuration modes)** command. This command option is available in all configuration modes except global configuration.

Example

- Type **show active** to display the content of ***running-config*** that affects the current mode. To include default settings in the display, type **show active all**.

```
Switch(config-router-ospf3)#show active all
ipv6 router ospf 9
   router-id 0.0.0.0
   default-metric 10
   distance ospf intra-area 10
   area 0.0.0.200 default-cost 10
   area 0.0.0.200
   no log-adjacency-changes
   timers spf 5
Switch(config-router-ospf3)#
```

To display any comments associated with the current mode, use the **comment** option of the **show (various configuration modes)** command.

Example

- Type **show comment** to display any comments attached to the current mode.

```
Switch(config-router-ospf3)#show comment
Comment for router-ospf3:
    Consult Thomas Morton before making changes to the OSPF configuration.
Switch(config-router-ospf3)#
```

3.5.2 Adding a Comment to a Configuration Mode

You can add a comment to many switch configuration modes by entering the **comment (various configuration modes)** command. Comments cannot be modified, but can be replaced by entering the **comment** command again and entering new text.

To append to an existing comment, enter ! followed by additional comment text. To display comments for the active mode, use the **comment** option of the **show (various configuration modes)** command. The **no comment** and **default comment** commands remove the comment from ***running-config***.

Comments cannot be added to the global configuration mode through the EOS, but can be added to all other configuration modes.

Example

- To add a comment to the active configuration mode, enter the **comment** command, then type the comment text. To end comment editing, type EOF on its own line (case sensitive) and press **enter**.

```
Switch(config-router-ospf3)#comment
Enter TEXT message. Type 'EOF' on its own line to end.
Consult Thomas Morton before making changes to the OSPF configuration.
EOF
Switch(config-router-ospf3)#
```

- To append to an existing comment, enter **!** followed by additional comment text.

```
Switch(config-router-ospf3)#!x2735
Switch(config-router-ospf3)#show comment
Comment for router-ospf3:
    Consult Thomas Morton before making changes to the OSPF configuration.
    x2735
Switch(config-router-ospf3)#
```

3.5.3 Saving the Running Configuration Settings

startup-config is the file, stored in internal flash memory, that the switch loads when it boots. Configuration changes that are not saved to ***startup-config*** are lost the next time the switch is booted.

The **write memory** and **copy running-config startup-config** commands store the operating configuration to ***startup-config.*** Both commands are supported in Privileged EXEC mode.

Example

- These equivalent commands save the current operating configure to the startup-config file.

```
Switch#write memory

Switch#copy running-config startup-config
```

The **show startup-config** command displays the startup configuration file. The command is supported in Privileged EXEC mode.

Example

- Type **show startup-config** to display the startup configuration file. The response in the example is truncated to display only the ip route configured in Admin Username on page 36.

```
Switch#show startup-config
! device: Switch (DCS-7124S, EOS-4.6.0-227198.EOS45)
!
<-------OUTPUT OMITTED FROM EXAMPLE-------->
!
ip route 0.0.0.0/0 192.0.2.1
!
<-------OUTPUT OMITTED FROM EXAMPLE-------->
!
end
Switch#
```

3.5.4 VRFs

An IP routing table is a data table that lists the routes to network destinations metrics (distances) associated with those routes. A routing table is also known as a routing information base (RIB).

VPN Routing and Forwarding (VRF) allows traffic separation by maintaining multiple routing tables. Arista switches support two VRF instances: a global or default VRF called “main” and a user defined VRF. The user defined VRF can be used as a management VRF to segregate management traffic from customer traffic. Routing cannot be performed within a user defined VRF.

Many commands that reference or modify the routing table allow the specification of a user defined VRF. Other commands reference a context active VRF, whose default routing table is a user defined VRF.

- Section 10.4.3 describes routing tables and VRFs
- Section 10.5.5 describes the configuration of VRFs.
- Section 10.5.5.3 describes context active VRFs.

3.6 Other Command-Line Interfaces

EOS can access other CLIs that provide switch commands, files, and services. .

- Section 3.6.1: Aboot Command-Line Interface describes the boot-loader CLI
- Section 3.6.2: Bash Shell describes the Bash shell CLI.

3.6.1 Aboot Command-Line Interface

Aboot is the switch boot loader. It reads a configuration file from the internal flash or a USB flash drive and attempts to boot a software image. The switch opens an Aboot shell if the switch does not find a software image, the configuration is corrupted, or the user terminates the boot process. The Aboot shell provides a CLI for manually booting a software image, recovering the internal flash to its default factory state, running hardware diagnostics, and managing files.

See Section 6.1: Boot Loader – Aboot for more information about Aboot.

3.6.2 Bash Shell

The switch provides a Linux Bash shell for accessing the underlying Linux operating system and extensions. The Bash shell is accessible in all command modes except EXEC. Section 3.4.1: Mode Types describes EOC command modes.

- To enter the Bash, type **bash** at the prompt.

```
Switch#bash

Arista Networks EOS shell

[admin@Switch ~]$
```

- To exit the Bash, type **logout, exit,** or **Ctrl-D** at the Bash prompt.

```
[admin@Switch ~]$ logout
Switch#
```

3.7 Directory Structure

EOS operates from a flash drive root mounted as the **/mnt/flash** directory on the switch. The EOS CLI supports these file and directory commands:

- **delete:** Delete a file or directory tree.
- **copy:** Copy a file.
- **more:** Display the file contents.
- **diff:** Compares the contents of files located at specified URLs.
- **rename:** Rename a file
- **cd:** Change the current working directory.
- **dir:** Lists directory contents, including files and subdirectories.
- **mkdir:** Create a directory.
- **rmdir:** Remove a directory.
- **pwd:** Display the current working directory.

Switch directory files are accessible through the Bash shell and Aboot. When entering the Bash shell from the switch, the working directory is located in **/home** directory and has the name of the user name from where Bash was entered.

Example

- These commands were entered from the user name john:

```
Switch#bash
[john@7124s ~]$ pwd
/home/john
[john@7124s ~]$
```

In this instance, the working directory is **/home/john**

When a flash drive is inserted in the USB flash port (see Figure 2-1), flash drive contents are accessible through **/mnt/usb1**.

When entering Aboot, the working directory is the root directory of the boot.

3.8 Command-Line Interface Commands

This section contains descriptions of the CLI commands that this chapter references.

Mode Navigation Commands

- alias . . . . . Page 70
- bash . . . . . Page 71
- configure (configure terminal) . . . . . Page 73
- daemon . . . . . Page 76
- enable . . . . . Page 78
- end . . . . . Page 79
- exit (Global Configuration) . . . . . Page 81

File Transfer Commands

- ip ftp client source-interface . . . . . Page 82
- ip http client source-interface . . . . . Page 83
- ip ssh client source-interface . . . . . Page 84
- ip tftp client source-interface . . . . . Page 85

File Commands

- copy running-config . . . . . Page 75
- configure network . . . . . Page 74

CLI Scheduling Commands

- schedule . . . . . Page 86
- show schedule . . . . . Page 89
- show schedule summary . . . . . Page 90

Event Handler Commands

- action bash . . . . . Page 69
- delay . . . . . Page 77
- event-handler . . . . . Page 80
- show event-handler . . . . . Page 88
- trigger . . . . . Page 93

Terminal Parameter Commands

- terminal length . . . . . Page 91
- terminal monitor . . . . . Page 92

Display and Comment Commands

- comment (various configuration modes) . . . . . Page 72
- show (various configuration modes) . . . . . Page 87
- show event-handler . . . . . Page 88
- show schedule . . . . . Page 89
- show schedule summary . . . . . Page 90

action bash

The **action bash** command specifies a Bash shell command to be run when an event handler is triggered. When an event handler is triggered, execution of the associated shell command is delayed by a configurable period set by the **delay** command. Only a single Bash command may be configured for an event handler, but the command may have multiple arguments. If more than one Bash command must be executed in response to a trigger, create a script containing the desired commands and enter the file path to the script as the argument of the **action bash** command.

To specify the event that will trigger the action, use the **trigger** command.

If the event handler uses an ***onIntf*** trigger, the following environment variables are passed to the action and can be used as arguments to the Bash command:

$INTF interface name.

$OPERSTATE current operational status of the specified interface.

$IP-PRIMARY current primary IP address of the specified interface.

Command Mode

Event-Handler Configuration

Command Syntax

action bash *command*

Parameters

- *command* Bash shell command to be executed when the event handler is triggered.

Example

- This command configures the event handler "onStartup" to run a script on the flash drive.

```
switch(config-handler-onStartup)#action bash /mnt/flash/myScript1
switch(config-handler-onStartup)#
```

- This command configures the event handler "eth_4" to send email to the specified address when there is a change in the operational status of Ethernet interface 4.

```
switch(config-event-eth_4)#action bash email x@yz.com -s "Et4 $OPERSTATE"
switch(config-event-eth_4)#
```

The above action uses the ***$OPERSTATE*** variable to include the current operational state ("linkup" or "linkdown") in the subject of the email. Note that the action will only function if email has been configured on the switch.

alias

The **alias** command creates an alias for a CLI command. Entering the alias in the CLI executes the corresponding command. Once created, an alias is accessible in all modes and all user sessions, but is subject to all the restrictions of the original command.

When using a command alias, no tokens may precede the alias except the **no** and **default** keywords. However, an alias can incorporate positional parameters.

In online help, aliases are indicated by an asterisk (*) and displayed in the following format:

**command_alias=original_command*

The **no alias** and **default alias** commands remove the specified alias. Preceding the alias itself with **no** executes the ***no*** form of the original command.

Command Mode

Global Configuration

Command Syntax

```
alias command_alias original_command
no alias command_alias
default alias command_alias
```

Parameters

- *command_alias* the string which is to be substituted for the original command. The string can include letters, numbers, and punctuation, but no spaces. If the ***command_alias*** string is identical to an existing command, the alias will supercede the original command.
- *original_command* the command which is to be executed when the alias is entered in the CLI. If the original command requires additional parameters, they must be included in the ***original_command*** string in the following manner:

  Positional parameters are of the form "%n" and must be whitespace-delimited. The first parameter is represented by "%1" and any additional parameters must be numbered sequentially. When executing the alias a value must be entered for each parameter or the CLI will display the error "% incomplete command".

Examples

- This command makes **e** an alias for the command **enable**

```
switch(config)#alias e enable
```

- This command makes **srie** an alias for the command **show running-config interface ethernet 1-6**

```
switch(config)#alias srie show running-config interface ethernet 1-6
```

- These commands make **ss** an alias for the command **show interfaces ethernet** <*range*> **status** with a positional parameter for the port range, then use the alias to display the status of ports 4/1-4/5

```
switch(config)#alias ss show interfaces ethernet %1 status
switch(config)#ss 4/1-4/5
Port      Name            Status       Vlan       Duplex  Speed Type
Et4/1                     connected    in Po1       full  10000 10GBASE-SRL
Et4/2                     notconnect   in Po1       full  10000 10GBASE-SRL
Et4/3                     notconnect   1            full  10000 10GBASE-SRL
Et4/4                     notconnect   1            full  10000 10GBASE-SRL
Et4/5                     notconnect   1            full  10000 10GBASE-SRL
```

bash

The **bash** command starts the Linux Bash shell. The Bash shell gives you access to the underlying Linux operating system and system extensions.

To exit the Bash, type **logout**, **exit**, or Ctrl-D at the Bash prompt.

Command Mode

Privileged EXEC

Command Syntax

bash

Examples

- This command starts the Bash shell.

```
switch#bash

Arista Networks EOS shell

[admin@switch ~]$
```

- This command, executed within Bash, exits the Bash shell.

```
[admin@switch ~]$ logout
switch#
```

comment (various configuration modes)

The **comment** command adds a comment for the active configuration mode to ***running-config***. Comments cannot be modified, but can be replaced by entering the **comment** command again and entering new text. To append to an existing comment, enter **!** followed by additional comment text. To display comments, use the **comment** option of the **show (various configuration modes)** command.

The **no comment** and **default comment** commands remove the comment from ***running-config***.

Comments cannot be added to the global configuration mode through the EOS..

Command Modes

All configuration modes except Global Configuration.

Command Syntax

```
comment comment_text EOF
no comment
default comment
! comment_text
```

Parameters

- *comment_text* To create a comment, enter a message when prompted. The message may span multiple lines.
- **EOF** To append to an existing comment, enter **!** followed by additional comment text.To end comment editing, type EOF on its own line (case sensitive) and press **enter**.

Example

- This command adds a comment to the active configuration mode.

```
switch(config-sg-radius-RAD-SV1)#comment
Enter TEXT message. Type 'EOF' on its own line to end.
Consult Thomas Morton before making changes to the RADIUS configuration.
EOF
switch(config-sg-radius-RAD-SV1)#
```

- This command appends a line to the comment for the active configuration mode.

```
switch(config-sg-radius-RAD-SV1)#! x3452
switch(config-sg-radius-RAD-SV1)#
```

configure (configure terminal)

The **configure** command places the switch in Global Configuration mode to configure features that affect the entire system. This mode also provides access to Interface Configuration mode and protocol-specific modes. The command may also be entered as **configure terminal**.

The **configure network** command refers the user to Arista's **copy** <*url*> **running-config** command for configuring the switch from a local file or network location.

Command Mode

Privileged EXEC

Command Syntax

```
configure
configure terminal
```

Example

- These commands place the switch in Global Configuration mode.

```
switch>enable
switch#configure
switch(config)#
```

configure network

The **configure network** command refers the user to Arista's **copy** <*url*> **running-config** command for configuring the switch from a local file or network location.

Command Mode

Privileged EXEC

Command Syntax

configure network

Example

- This is the output of the **configure network** command.

```
switch#configure network
%% Please use copy <url> running-config
switch#
```

copy running-config

The current operating configuration of the switch is stored in a virtual file called ***running-config***. The **copy running-config** command saves the contents of the ***running-config*** virtual file to a new location.

Command Mode

Privileged EXEC

Command Syntax

```
copy running-config DESTINATION
```

Parameters

- ***DESTINATION*** destination for the contents of the ***running-config*** file. Values include:
  - **startup-config** the configuration file that the switch loads when it boots.

    The **copy running-config startup-config** and **write memory** commands are equivalent.
  - file: a file in the switch file directory.
  - flash: a file in flash memory.
  - url any valid URL.

    The **copy running-config url** and **write network url** commands are equivalent.

Examples

- This command copies ***running-config*** to the ***startup-config*** file.

```
switch#copy running-config startup-config
switch#
```

- This command copies ***running-config*** to a file called rc20110617 in the dev subdirectory of the switch directory.

```
switch#copy running-config file:dev/rc20110617
switch#
```

daemon

The **daemon** command accesses daemon configuration mode for adding or removing external daemons and scripts, which are then managed by ProcMgr.

The **no daemon** and **default daemon** commands delete the deamon by removing the corresponding **daemon** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

daemon *daemon_name*

Parameters

- *daemon_name* label that references the daemon configuration mode.

Examples

- These commands enters daemon configuration mode and initiates the daemon script.

```
switch(config)#daemon process1
switch(config-daemon-process1)#command process-script -i -m
switch(config-daemon-process1)#
```

delay

The **delay** command specifies the time in seconds the system will delay between a triggering event and the execution of an event handler action. The default delay is 20 seconds.

Command Mode

Event-Handler Configuration

Command Syntax

delay *seconds*

Parameters

- *seconds* number of seconds to delay before executing the action. The default is 20.

Example

- This command configures the event handler Eth5 to delay 10 seconds before executing.

```
switch(config-handler-Eth5)#delay 10
switch(config-handler-Eth5)#
```

enable

The **enable** command places the switch in Privileged EXEC mode. If an ***enable*** password is set, the CLI displays a password prompt when a user enters the **enable** command. If the user enters an incorrect password three times, the CLI displays the EXEC mode prompt.

To set a local ***enable*** password, use the **enable secret** command.

Command Mode

EXEC

Command Syntax

enable [*privilege_level*]

Parameters

- *privilege_level* optional privilege level for this session. Values range from 0 to 15; the default is 15. Any level above 1 is Privileged EXEC mode. Setting the ***privilege_level*** to 0 or 1 leaves the switch in EXEC mode.

Example

- This command places the switch in Privileged EXEC mode with the default privilege level of 15.

```
switch>enable
switch#
```

end

The **end** command exits to Privileged Exec mode from any Configuration mode. If the switch is in a group-change mode (such as ACL-Configuration mode or MST-Configuration mode), the **end** command also saves all pending changes made in that mode to ***running-config***.

Command Mode

any Configuration mode

Command Syntax

end

Example

- This command exits to Privileged Exec mode.

```
switch(config-if-Et25)#end
switch#
```

event-handler

An event handler executes a Linux Bash shell command in response to a specific system event. An event handler consists of a Bash command, a trigger and a delay; when the trigger event occurs, the action is scheduled to run after ***delay*** seconds.

The **event-handler** command places the switch in event-handler configuration mode for the specified event handler. If the named event handler does not already exist, this command creates it. Event-handler configuration mode is a group change mode that configures event handlers.

Changes made in a group change mode are saved by leaving the mode through the **exit** command or by entering another configuration mode.

These commands are available in event-handler configuration mode:

- **action bash**
- **delay**
- **trigger**

The **no event-handler** and **default event-handler** commands delete the specified event handler by removing it from ***running config***.

Command Mode

Global Configuration

Command Syntax

```
event-handler name
no event-handler name
default event-handler name
```

Parameters

- *name* name of the event handler to be configured. If the named event handler does not already exist, this command will create it.

Example

- This command places the switch in event-handler configuration mode for an event handler called "Eth_5".

```
switch(config)#event-handler Eth_5
switch(config-handler-Eth_5)#
```

exit (Global Configuration)

The **exit** command exits global configuration mode to Privileged EXEC mode. If used in EXEC or Privileged EXEC mode, the **exit** command terminates the user session. If the switch is in a group-change mode (such as ACL-Configuration mode or MST-Configuration mode), the **exit** command will also apply any pending changes made in that mode.

Command Mode

Global Configuration

Command Syntax

exit

Example

- This command exits Global Configuration mode to Privileged EXEC mode.

```
switch(config)#exit
switch#
```

- This command terminates the user session.

```
switch#exit
```

ip ftp client source-interface

The **ip ftp client source-interface** command specifies the source IP address for File Transfer Protocol (FTP) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The **ip ftp client source-interface** and **ip ftp source-interface** commands are functionally equivalent. In each case, **ip ftp client source-interface** is stored in ***running-config***

The **no ip ftp client source-interface** and **default ip ftp client source-interface** commands restore default behavior by removing the **ip ftp client source-interface** from ***running-config*.**

Command Mode

Global Configuration

Command Syntax

```
ip ftp [client] source-interface INTERFACE
no ip ftp [client] source-interface
default ip ftp [client] source-interface
```

Parameters

- **client** Parameter has no functional effect.
- *INTERFACE* Interface providing the IP address. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- These commands configure the 10.10.121.15 as the source IP address the switch uses when communicating with FTP servers.

```
switch(config)#interface ethernet 17
switch(config-if-Et17)#ip address 10.10.121.15/24
! IP configuration will be ignored while interface Ethernet17 is not a routed
port.
switch(config-if-Et17)#ip ftp client source-interface ethernet 17
switch(config-if-Et17)#
```

ip http client source-interface

The **ip http client source-interface** command specifies the source IP address for hypertext transfer protocol (HTTP) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The **no ip http client source-interface** and **default ip http client source-interface** commands restore default behavior by removing the **ip http client source-interface** from ***running-config*.**

Command Mode

Global Configuration

Command Syntax

```
ip http client source-interface INTERFACE
no ip http client source-interface
default ip http client source-interface
```

Parameters

- *INTERFACE* Interface providing the IP address. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- These commands configure the 10.15.17.9 as the source IP address the switch uses when communicating with http servers.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.15.17.9/24
switch(config-if-Vl10)#ip http client source-interface vlan 10
switch(config)#
```

ip ssh client source-interface

The **ip ssh client source-interface** command specifies the source IP address for secure shell (SSH) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The **ip ssh client source-interface** and **ip ssh source-interface** commands are functionally equivalent. In each case, **ip ssh client source-interface** is stored in ***running-config***

The **no ip ssh client source-interface** and **default ip ssh client source-interface** commands restore default behavior by removing the **ip ssh client source-interface** from ***running-config*.**

Command Mode

Global Configuration

Command Syntax

```
ip ssh [client] source-interface INTERFACE
no ip ssh [client] source-interface
default ip ssh [client] source-interface
```

Parameters

- **client** Parameter has no functional effect.
- *INTERFACE* Interface providing the IP address. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- These commands configure the 10.17.17.9 as the source IP address the switch uses when communicating with ssh servers.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.15.17.9/24
switch(config-if-Vl10)#ip ssh client source-interface vlan 10
switch(config)#
```

ip tftp client source-interface

The **ip tftp client source-interface** command specifies the source IP address for Trivial File Transfer Protocol (TFTP) connections. By default, the source IP address is selected by the switch when this command is not configured or when the specified interface is not assigned an IP address.

The **ip tftp client source-interface** and **ip tftp source-interface** commands are functionally equivalent. In each case, **ip tftp client source-interface** is stored in ***running-config***

The **no ip tftp client source-interface** and **default ip tftp client source-interface** commands restore default behavior by removing the **ip tftp client source-interface** from ***running-config*.**

Command Mode

Global Configuration

Command Syntax

```
ip tftp [client] source-interface INTERFACE
no ip tftp [client] source-interface
default ip tftp [client] source-interface
```

Parameters

- **client** Parameter has no functional effect.
- *INTERFACE* Interface providing the IP address. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- These commands configure the 10.15.17.9 as the source IP address the switch uses when communicating with tftp servers.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip address 10.15.17.9/24
switch(config-if-Vl10)#ip tftp client source-interface vlan 10
switch(config)#
```

schedule

The **schedule** command facilitates the periodic execution of a specified CLI command. Command parameters configure the interval between consecutive execution instances and the maximum number of files that can be created when the command requires log files. By default, periodic execution of the following ***show tech-support*** command is enabled:

```
schedule tech-support interval 60 max-log-files 100 command show tech-support
```

Text that the CLI normally displays as a result of executing the scheduled command through the CLI is stored in log files at ***flash:/schedule/<sched_name>***. Empty log files are created for commands that do not generate CLI text.

The **no schedule** and **default schedule** commands disable execution of the specified command by removing the corresponding **schedule** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
schedule sched_name interval period max-log-files num_files command cli_name
no schedule sched_name
default schedule sched_name
```

Parameters

- *sched_name* label associated with the scheduled command.
- *period* period between consecutive execution iterations (minutes). Value ranges from 1 to 1440.
- *num_files* maximum number of log files command generates for std output. Range is 1 to 10000.
- *cli_name* name of the CLI command.

Guidelines

Log files created by the command are stored in the **flash:/schedule/<*sched_name*>/** directory.

Examples

- This command schedules the execution of a script file once every 12 hours. The log file option is set to the option minimum of one because the command does not generate output to the CLI.

```
switch(config)#schedule ms_1 interval 720 max-log-files 1 command bash
/mnt/flash/myscript.sh
```

This command displays the commands that are scheduled for periodic execution.

```
switch(config)#show schedule summary
Name             Last   Interval Max log   Log file location
                 time   (mins)   files
---------------- -----  -------- --------  -----------------
tech-support     16:13   60      100       flash:/schedule/tech-support
ms_1             16:28   720     1         flash:/schedule/ms_1
```

- This command stores ***running-config*** contents to a log file once each hour, creating up to 24 log files.

```
switch(config)#schedule backup-test interval 60 max-log-files 24 command show
running-config
```

show (various configuration modes)

The **show** command, when executed within a configuration mode, can display data in ***running-config*** for the active configuration mode.

Command Modes

All configuration modes except Global Configuration

Command Syntax

show [***DATA_TYPE***]

Parameters

- *DATA_TYPE* Specifies display contents. Values include:
  - **active** Displays ***running-config*** settings for the configuration mode.
  - **active all** Displays ***running-config*** plus defaults for the configuration mode.
  - **active all detail** Displays ***running-config*** plus defaults for the configuration mode.
  - **comment** Displays comment entered for the configuration mode.

Related Commands

The **show** commands in ACL-configuration mode and MST-configuration mode include the **active** and **comment** options along with additional mode-specific options.

Examples

- This command shows the server-group-TACACS+ configuration commands in ***running-config.***

```
switch(config-sg-tacacs+-TAC-GR)#show active
   server TAC-1
   server 10.1.4.14
switch(config-sg-tacacs+-TAC-GR)#
```

show event-handler

The **show event-handler** command displays the contents and activation history of a specified event handler or all event handlers.

Command Mode

Privileged EXEC

Command Syntax

show event-handler [*handler_name*]

Parameters

- *handler_name* optional name of an event handler to display. If no parameter is entered, the command displays information for all event handlers configured on the system.

Example

- This command displays information about an event handler called "eth_5".

```
switch#show event-handler eth_5
Event-handler eth_5
Trigger: onIntf Ethernet5 on operstatus delay 20 seconds
Action: /mnt/flash/myScript1
Last Trigger Activation Time: Never
Total Trigger Activations: 0
Last Action Time: Never
Total Actions: 0
switch#
```

show schedule

The **show schedule** command displays logging output on the terminal during the current terminal session. This command affects only the local monitor. The **no terminal monitor** command disables direct monitor display of logging output for the current terminal session.

Command Mode

Privileged EXEC

Command Syntax

show schedule *schedule_name*

Parameters

- *schedule_name* label associated with the scheduled command.

Example

- This command displays logging to the local monitor during the current terminal session.

```
switch#show schedule tech-support
CLI command "show tech-support" is scheduled, interval is 60 minutes
Maximum of 100 log files will be stored
100 log files currently stored in flash:/schedule/tech-support

Start Time           Size    Filename
-------------------  -----   --------
Jan 19 2011 00:00    14 kB   tech-support_2011-01-19.0000.log.gz
Jan 19 2011 04:00    14 kB   tech-support_2011-01-19.0100.log.gz
...
```

show schedule summary

The **show schedule summary** command displays the list of active scheduled commands.

Command Mode

Privileged EXEC

Command Syntax

```
show schedule summary
```

Example

- This command displays the list of active scheduled commands.

```
switch#show schedule summary
Name           Last   Interval Max log   Log file location
               time   (mins)   files
-------------- ------  -------  --------  ----------------------------------
tech-support   00:00    60     100       flash:/schedule/tech-support
Et45-counters  00:05     5     100       flash:/schedule/Et45-counters
Memfree        00:10    10     100       flash:/schedule/Memfree
```

terminal length

The **terminal length** command overrides automatic pagination and sets pagination length for all show commands on a terminal. If the output of a show command is longer than the configured terminal length, the output will be paused after each screenful of output, prompting the user to continue.

To disable pagination for an SSH session, set ***terminal length*** to 0. By default, all console sessions have pagination disabled.

The **no terminal length** command restores automatic pagination by removing the **terminal length** command from ***running-config***.

The pagination setting is persistent if configured from Global Configuration mode. If configured from EXEC mode, the setting applies only to the current CLI session. Pagination settings may also be overridden when you adjust the size of the SSH terminal window, but can be reconfigured by running the **terminal length** command again.

Command Mode

EXEC

Command Syntax

```
terminal length lines
no terminal length
```

Parameters

- *lines* number of lines to be displayed at a time. Values range from 0 through 32767. A value of 0 disables pagination.

Example

- This command sets the pagination length for the current terminal session to 10 lines.

```
switch#terminal length 10
Pagination set to 10 lines.
```

- This command configures the switch to paginate terminal output automatically based on screen size for the current terminal session.

```
switch#no terminal length
```

- These commands disable pagination globally.

```
switch#configure
switch(config)#terminal length 0
Pagination disabled.
```

terminal monitor

The **terminal monitor** command enables the display of logging output on the terminal during the current terminal session. This command affects only the local monitor. The **no terminal monitor** command disables direct monitor display of logging output for the current terminal session.

Command Mode

Privileged EXEC

Command Syntax

```
terminal monitor
no terminal monitor
default terminal monitor
```

Example

- This command enables the display of logging to the local monitor during the current terminal session.

```
switch#terminal monitor
switch#
```

trigger

The **trigger** command specifies what event will trigger the event handler. Handlers can be triggered either by the system booting or by a change in a specified interface's IP address or operational status.

To specify the action to be taken when a triggering event occurs, use the **action bash** command.

Command Mode

Event-Handler Configuration

Command Syntax

```
trigger EVENT
```

Parameters

- *EVENT* event which will trigger the configuration mode event handler. Values include:
  - **onboot** triggers when the system reboots, or when you exit event-handler configuration mode. This option takes no further arguments, and passes no environment variables to the action triggered.
  - **onintf** *INTERFACE CHANGE* triggers when a change is made to the specified interface.
- *INTERFACE* the triggering interface. Values include:
  - **ethernet** *number* Ethernet interface specified by *number*.
  - **loopback** *number* loopback interface specified by *number*.
  - **management** *number* management interface specified by *number*.
  - **port-channel** *number* channel group interface specified by *number*.
  - **vlan** *numver* VLAN interface specified by *number*.
- *CHANGE* the change being watched for in the triggering interface. Values include:
  - **ip** triggers when the IP address of the specified interface is changed.
  - **operstatus** triggers when the operational status of the specified interface changes.

Examples

- This command configures the event handler "Eth5" to be triggered when there is a change in the operational status or IP address of Ethernet interface 5.

```
switch(config-handler-Eth5)#trigger onIntf Ethernet 5 operstatus ip
switch(config-handler-Eth5)#
```

- This command configures the event handler "onStartup" to be triggered when the system boots, or on exiting event-handler configuration mode.

```
switch(config-handler-onStartup)#trigger onboot
switch(config-handler-onStartup)#
```

Chapter 4

AAA Configuration

This chapter describes authentication, authorization, and accounting configuration tasks and contains these sections:

- Section 4.1: Authorization, Authentication, and Accounting Overview
- Section 4.2: Configuring the Security Services
- Section 4.3: Activating Security Services
- Section 4.4: Security Configuration Examples
- Section 4.5: AAA Commands

4.1 Authorization, Authentication, and Accounting Overview

4.1.1 Methods

The switch controls access to EOS commands by authenticating user identity and verifying user authorization. Authentication, authorization, and accounting activities are conducted through three data services – a local security database, TACACS+ servers, and RADIUS servers. Section 4.2: Configuring the Security Services describes these services.

4.1.2 Configuration Statements

Switch security requires two steps:

1. Configuring security service parameters.

   EOS provides configuration commands for each security service:

   - A local file supports authentication through **username** and **enable secret** commands.
   - TACACS+ servers provide security services through **tacacs-server** commands.
   - RADIUS servers provide security services through **radius-server** commands.

   Section 4.2: Configuring the Security Services describes security service configuration commands.

2. Activating authentication, authorization, and accounting services.

   EOS provides **aaa authorization**, **aaa authentication**, and **aaa accounting** commands to select the primary and backup services. Section 4.3: Activating Security Services provides information on implementing a security environment.

4.1.3 Encryption

EOS uses clear text passwords and server access keys to authenticate users and communicate with security systems. To prevent accidental disclosure of these passwords and keys, EOS stores their corresponding encrypted strings. The encryption method depends on the type of password or key.

EOS commands that configure passwords or keys can accept the clear text password or an encrypted string that was generated by the specified encryption algorithm with the clear text password as the seed.

4.2 Configuring the Security Services

EOS can access three security data services when authenticating users and authorizing switch tasks: a local file, TACACS+ servers, and RADIUS Servers.

4.2.1 Local

The local file uses passwords to provide these authentication services:

- authenticate users as they log into the switch
- control access to configuration commands
- control access to the switch root login

The local file contains username-password combinations to authenticate users. Passwords also authorize access to configuration commands and the switch root login.

4.2.1.1 Passwords

The switch recognizes passwords in their forms as clear text and encrypted strings.

- **Clear text** passwords is the text that the a user enters to access the CLI, configuration commands, or the switch root login.
- **Encrypted strings** are MD5-encrypted strings generated with the **clear text** as the seed. The local file stores passwords in this format to avoid unauthorized disclosure. When a user enters the clear text password, the switch generates the corresponding secure hash and compares it to the stored version. **The switch cannot recover the clear text from which an encrypted string is generated**.

Valid passwords contain the characters A-Z, a-z, 0-9 and any of these punctuation characters:

```
!   @   #   $   %   ^   &   *   (   )   -   _   =   +
{   }   [   ]   ;   :   <   >   ,   .   ?   /   ~   \
```

4.2.1.2 Usernames

Usernames control access to the EOS and all switch commands. The switch is typically accessed through an SSH login, using a previously defined username-password combination. To create a new username or modify an existing username, use the **username** command.

Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

```
@   #   $   %   ^   &   *   (   )   -   _   =
+   {   }   [   ]   ;   <   >   ,   .   ~   |
```

The default username is **admin,** which is described in Admin Username.

Examples

- These equivalent commands create the username *john* and assign it the password *x245*. The password is entered in clear text because the encrypt-type parameter is omitted or zero.

```
Switch(config)#username john secret x245

Switch(config)#username john secret 0 x245
```

- This command creates the username *john* and assigns it to the text password that corresponds to the encrypted string *1sU.7hptc$TsJ1qslCL7ZYVbyXNG1wg1*. The string was generated by an MD5-encryption program using x245 as the seed.

```
Switch(config)#username john secret 5 $1$sU.7hptc$TsJ1qslCL7ZYVbyXNG1wg1
```

  The username is authenticated by entering ***x245*** when the CLI prompts for a password.

- This command creates the username *jane* without securing it with a password. It also removes a password if the *jane* username exists.

```
Switch(config)#username jane nopassword
```

- This command removes the username **william** from the local file.

```
Switch(config)#no username william
```

4.2.1.3 Logins by Unprotected Usernames

The default switch configuration allows usernames that are not password protected to log in only from the console. The **aaa authentication policy local** command configures the switch to allow unprotected usernames to log in from any port. To reverse this setting to the default state, use **no aaa authentication policy local allow-nopassword-remote-login.**

Warning Allowing remote access to accounts without passwords is a severe security risk. Arista Networks recommends assigning strong passwords to all usernames.

Examples

- This command configures the switch to allow unprotected usernames to login from any port.

```
S(config)#aaa authentication policy local allow-nopassword-remote-login
S(config)#
```

- This command configures the switch to allow unprotected usernames to login only from the console port.

```
S(config)#no aaa authentication policy local allow-nopassword-remote-login
S(config)#
```

4.2.1.4 Enable Command Authentication

The **enable** command controls access to Privileged EXEC and all configuration command modes. The enable password authorizes users to execute the **enable** command. When the enable password is set, the CLI displays a password prompt when a user attempts to enter Privileged EXEC mode.

```
main-host>enable
Password:
main-host#
```

If the user enters an incorrect password three times, the CLI displays the EXEC mode prompt.

If the enable password is not set, the CLI does not prompt for a password when a user attempts to enter Privileged EXEC mode.

To set the enable password, use the **enable secret** command.

Examples

- These equivalent commands assign *xyrt1* as the enable password.

```
Switch(config)#enable secret xyrt1

Switch(config)#enable secret 0 xyrt1
```

- This command assigns the enable password to the clear text (*12345*) corresponding to the encrypted string *$1$8bPBrJnd$Z8wbKLHpJEd7d4tc5Z/6h/*. The string was generated by an MD5-encryption program using *12345* as the seed.

```
Switch(config)#enable secret 5 $1$8bPBrJnd$Z8wbKLHpJEd7d4tc5Z/6h/
```

- This command deletes the enable password.

```
Switch(config)#no enable secret
```

4.2.1.5 Root Account Password

The root account accesses the root directory in the underlying Linux shell. When it is not password protected, you can log into the root account only through the console port. After you assign a password to the root account, you can log into it through any port.

To set the password for the root account, use the **aaa root** command.

Examples

- These equivalent commands assign *f4980* as the root account password.

```
Switch(config)#aaa root secret f4980

Switch(config)#aaa root secret 0 f4980
```

- This command assigns the text (*ab234*) that corresponds to the encrypted string of *1HW05LEY8$QEVw6JqjD9VqDfh.O8r.b.* as the root password.

```
Switch(config)#aaa root secret 5 $1$HW05LEY8$QEVw6JqjD9VqDfh.O8r.b
```

- This command removes the password from the root account.

```
Switch(config)#aaa root nopassword
```

- This command disables the root login.

```
Switch(config)#no aaa root
```

4.2.2 TACACS+

Terminal Access Controller Access-Control System Plus (TACACS+) is a security system that provides centralized user validation services. TACACS+ information is maintained on a remote database. EOS support of TACACS+ services requires access to a TACACS+ server.

TACACS+ manages multiple network access points from a single server. A network access server provides connections to a single user, to a network or subnetwork, and to interconnected networks.

The switch defines a TACACS+ server connection by its address and port. This allows the switch to conduct multiple data streams to a single server by addressing different ports on the server.

These sections describe steps that configure access to TACACS+ servers. Configuring TACACS+ access is most efficiently performed when TACACS+ is functioning prior to configuring switch parameters.

4.2.2.1 Configuring TACACS+ Parameters

TACACS+ parameters define settings for the switch to communicate with TACACS+ servers. A set of values can be configured for individual TACACS+ servers that the switch accesses. Global parameters define settings for communicating with servers for which parameters are not individually configured.

The switch supports these TACACS+ parameters:

Encryption key

The encryption key is code that switch and TACACS+ server shares to facilitate communications.

- The **tacacs-server host** command defines the encryption key for a specified server.
- The **tacacs-server key** command defines the global encryption key.

Examples

- This command configures the switch to communicate with the TACACS+ server assigned the host name *TAC_1* using the encryption key *rp31E2v*.

```
Switch(config)#tacacs-server host TAC_1 key rp31E2v
```

- This command configures *cv90jr1* as the global encryption key.

```
Switch(config)#tacacs-server key 0 cv90jr1
```

- This command assigns *cv90jr1* as the global key, using the corresponding encrypted string.

```
Switch(config)#tacacs-server key 7 020512025B0C1D70
```

Session Multiplexing

The switch supports multiplexing sessions on a single TCP connection.

- The **tacacs-server host** command configures the multiplexing option for a specified server.
- There is no global multiplexing setting.

Example

- This command configures the switch to communicate with the TACACS+ server at *10.12.7.9* and indicates the server supports session multiplexing on a TCP connection.

```
Switch(config)#tacacs-server host 10.12.7.9 single-connection
```

Timeout

The timeout is the period the switch waits for a successful connection to or response from the TACACS+ server. The default is 5 seconds.

- The **tacacs-server host** command defines the timeout for a specified server.
- The **tacacs-server timeout** command defines the global timeout.

Examples

- This command configures the switch to communicate with the TACACS+ server assigned the host name *TAC_1* and configures the timeout period as *20 seconds*.

```
Switch(config)#tacacs-server host TAC_1 timeout 20
```

- This command configures *40 seconds* as the period that the server waits for a response from a TACACS+ server before issuing an error.

```
Switch(config)#tacacs-server timeout 40
```

Port

The port specifies the port number through which the switch and the servers send information. The TACACS+ default port is 49.

- The **tacacs-server host** command specifies the port number for an individual TACACS+ server.
- The global TACACS+ port number cannot be changed from the default value of 49.

Example

- This command configures the switch to communicate with the TACACS+ server at *10.12.7.9* through port 54.

```
Switch(config)#tacacs-server host 10.12.7.9 port 54
```

4.2.2.2 TACACS+ Status

To display the TACACS+ servers and their interactions with the switch, use the **show tacacs** command.

Example

- This command lists the configured TACACS+ servers.

```
switch(config)#show tacacs

server1: 10.1.1.45
Connection opens: 15
Connection closes: 6
Connection disconnects: 6
Connection failures: 0
Connection timeouts: 2
Messages sent: 45
Messages received: 14
Receive errors: 2
Receive timeouts: 2
Send timeouts: 3

Last time counters were cleared: 0:07:02 ago
```

To reset the TACACS+ status counters, use the **clear aaa counters tacacs+** command.

Example

- This command clears all TACACS+ status counters.

```
Switch(config)#clear aaa counters tacacs
```

4.2.3 RADIUS

Remote Authentication Dial In User Service (RADIUS) is a networking protocol that provides centralized authentication, authorization, and accounting services for computers connecting to and using network resources. RADIUS is used to manage access to the Internet, internal networks, wireless networks, and integrated email services.

These sections describe steps that configure access to a RADIUS server. Configuring RADIUS parameters is most efficiently performed when RADIUS is functioning prior to configuring switch parameters.

4.2.3.1 Configuring RADIUS Defaults

RADIUS policies specify settings for the switch to communicate with RADIUS servers. A set of values can be configured for individual RADIUS servers that the switch accesses. Global parameters define settings for communicating with servers for which parameters are not individually configured.

The switch defines these RADIUS parameters:

Encryption key

The encryption key is the key shared by the switch and RADIUS servers to facilitate communications.

- The **radius-server host** command defines the encryption key for a specified server.
- The **radius-server key** command specifies the global encryption key.

Examples

- This command configures the switch to communicate with the RADIUS server assigned the host name *RAD-1* using the encryption key *rp31E2v*.

```
Switch(config)#radius-server host RAD-1 key rp31E2v
```

- This command configures *cv90jr1* as the global encryption key.

```
Switch(config)#radius-server key 0 cv90jr1
```

- This command assigns *cv90jr1* as the key by specifying the corresponding encrypted string.

```
Switch(config)#radius-server key 7 020512025B0C1D70
```

Timeout

The timeout is the period that the switch waits for a successful connection to or response from a RADIUS server. The default period is 5 seconds.

- The **radius-server host** command defines the timeout for a specified server.
- The **radius-server timeout** command defines the global timeout.

Examples

- This command configures the switch to communicate with the RADIUS server assigned the host name *RAD-1* and configures the timeout period as *20 seconds*.

```
Switch(config)#radius-server host RAD-1 timeout 20
```

- This command configures *50 seconds* as the period that the server waits for a response from a RADIUS server before issuing an error.

```
Switch(config)#radius-server timeout 50
```

Retransmit

Retransmit is the number of times the switch attempts to access the RADIUS server after the first server timeout expiry. The default value is 3 times.

- The **radius-server host** command defines the retransmit for a specified server.
- The **radius-server retransmit** command defines the global retransmit value.

Examples

- This command configures the switch to communicate with the RADIUS server assigned the host name *RAD-1* and configures the retransmit value as 2.

```
Switch(config)#radius-server host RAD-1 retransmit 2
```

- This command configures the switch to attempt five RADIUS server contacts after the initial timeout. If the timeout parameter is set to *50 seconds*, then the total period that the switch waits for a response is ((5+1)*50) = 300 seconds.

```
Switch(config)#radius-server retransmit 5
```

Deadtime

Deadtime is the period when the switch ignores a non-responsive RADIUS server, or a server that does not answer retransmit attempts after timeout expiry. Deadtime is disabled if a value is not specified.

- The **radius-server host** command defines the deadtime for a specified server.
- The **radius-server deadtime** command defines the global deadtime setting.

Examples

- This command configures the switch to communicate with the RADIUS server assigned the host name *RAD-1* and configures the deadtime period as 90 minutes.

```
Switch(config)#radius-server host RAD-1 deadtime 90
```

- This command programs the switch to ignore a server for two hours if the server does not respond to a request during the timeout-retransmit period.

```
Switch(config)#radius-server deadtime 120
```

Port

The port specifies the port number through which the switch and servers send information.

- The **radius-server host** command specifies the port numbers for an individual RADIUS server.
- The global RADIUS port numbers cannot be changed from the default values of 1812 for an authorization port and 1813 for an accounting port.

Example

- These commands configure the switch to communicate with the RADIUS server named *RAD-1* through port number 1850 for authorization and port number 1851 for accounting.

```
Switch(config)#radius-server host RAD-1 auth-port 1850
Switch(config)#radius-server host RAD-1 acct-port 1851
```

To remove the configuration for this server, use **no radius-server host** command and specify the hostname or IP address with both the authorization and accounting port numbers.

4.2.3.2 RADIUS Status

The **show radius** command displays configured RADIUS servers and their interactions with the switch.

Example

- This command lists the configured RADIUS servers.

```
Switch(config)#show radius

server1: 10.1.1.45
Messages sent: 24
Messages received: 20
Requests accepted: 14
Requests rejected: 8
Requests timeout: 2
Requests retransmitted: 1
Bad responses: 1
Last time counters were cleared: 0:07:02 ago
```

To reset the RADIUS status counters, use the **clear aaa counters radius** command.

Example

- This command clears all RADIUS status counters.

```
Switch(config)#clear aaa counters radius
```

4.2.4 Server Groups

A server group is a collection of servers that are associated with a single label. Subsequent authorization and authentication commands access all servers in a group by invoking the group name.

The switch supports TACACS+ and RADIUS server groups. Use the **aaa group server radius** or **aaa group server tacacs+** command to create a named server group. In addition to creating the server group, the CLI enters server group configuration mode for the specified group. Server group members must be previously configured with a **radius-server host** or **tacacs-server host** command.

Examples

- This command creates the TACACS+ server group named TAC-GR and enters server group configuration mode for the new group.

```
Switch(config)#aaa group server tacacs+ TAC-GR
Switch(config-sg-tacacs+-TAC-GR)#
```

- These commands add two servers to the TAC-GR server group. To add servers to this group, the switch must be in sg-tacacs+-TAC-GR configuration mode.

  The CLI remains in server group configuration mode after adding the TAC-1 server (port 49) and the server located at 10.1.4.14 (port 151) to the group.

```
Switch(config-sg-tacacs+-TAC-GR)#server TAC-1
Switch(config-sg-tacacs+-TAC-GR)#server 10.1.4.14 port 151
Switch(config-sg-tacacs+-TAC-GR)#
```

- This command exits server group mode.

```
Switch(config-sg-tacacs+-TAC-GR)#exit
Switch(config)#
```

- This command creates the RADIUS server group named RAD-SV1 and enters server group configuration mode for the new group.

```
Switch(config)#aaa group server radius RAD-SV1
Switch(config-sg-radius-RAD-SV1)#
```

- These commands add two servers to the RAD-SV1 server group. To add servers to this group, the switch must be in sg-radius-RAD-SV1 configuration mode.

  The CLI remains in server group configuration mode after adding the RAC-1 server (authorization port 1812, accounting port 1813) and the server located at 10.1.5.14 (authorization port 1812, accounting port 1850) to the group.

```
Switch(config-sg-radius-RAD-SV1)#server RAC-1
Switch(config-sg-radius-RAD-SV1)#server 10.1.5.14 acct-port 1850
Switch(config-sg-radius-RAD-SV1)#
```

4.3 Activating Security Services

After configuring the access databases, **aaa authentication** and **aaa authorization** commands designate active and backup services for handling access requests.

4.3.1 Service Lists

These sections describe the methods of selecting the database that the switch uses to authenticate users and authorize access to network resources.

Service lists specify the service by which the switch authenticates usernames and the enable password. List elements are service options, ordered by the priority that the switch attempts to use them.

Example

- This is an example service list for username authentication:

  1. Location_1 server group – specifies a server group (Section 4.2.4: Server Groups).
  2. Location_2 server group – specifies a server group (Section 4.2.4: Server Groups).
  3. TACACS+ servers – specifies all hosts for which a **tacacs-server host** command exists.
  4. Local file – specifies the local file
  5. None – specifies that no authentication is required – all access attempts succeed.

  To authenticate a username, the switch checks Location_1 server group. If a server in the group is available, the switch authenticates the username through that group. Otherwise, it continues through the list until it finds an available service or utilizes option 5, which allows the access attempt to succeed without authentication.

4.3.2 Authenticating Usernames and the Enable Password

These commands specify service lists that authenticate usernames and the enable command password:

- **aaa authentication login** specify services the switch uses to authenticates usernames.
- **aaa authentication enable** specify services the switch uses to authenticates the enable password.

Examples

- This command configures the switch to authenticate usernames through the TAC-1 server group. The local database is the backup method if TAC-1 servers are unavailable.

  ```
  Switch(config)#aaa authentication login default group TAC-1 local
  ```

- This command configures the switch to authenticate usernames through all TACACS+ servers, then all RADIUS servers if the TACACS+ servers are not available. If the RADIUS servers are unavailable, the switch does not authenticate any login attempts.

  ```
  Switch(config)#aaa authentication login default group tacacs+ group radius
  none
  ```

- This command configures the switch to authenticate the enable password through all TACACS+ servers, then through the local database if the TACACS+ servers are unavailable.

  ```
  Switch(config)#aaa authentication enable default group TACACS+ local
  ```

4.3.3 Authorization

Authorization commands control access to the EOS shell and CLI commands. Authorization also controls configuration access through the console port.

- To specify the method through which the switch authorizes opening a CLI shell, enter **aaa authorization exec**.
- To specify the method through which the switch authorizes commands, enter **aaa authorization commands**.

  During the exec authorization process, the TACACS+ server response may include attribute-value (AV) pairs. The switch recognizes **priv-lvl=x** (where **x** is an integer between 0 and 15), which is a mandatory AV pair. A TACACS+ server that sends any other mandatory AV pair is denied access to the switch. The receipt of optional AV pairs by the switch has no affect on decisions to permit or deny access to the TACACS+ server.

  Examples

  - This command specifies that TACACS+ servers authorize users attempting to open a CLI shell.

    ```
    switch(config)#aaa authorization exec default group tacacs+
    switch(config)#
    ```

  - This command programs the switch to authorize configuration commands (privilege level 15) through the local file and to deny command access to users not listed in the local file.

    ```
    switch(config)#aaa authorization commands 15 default local
    switch(config)#
    ```

  - This command programs the switch to permit all commands entered on the CLI.

    ```
    switch(config)#aaa authorization commands all default none
    switch(config)#
    ```

All commands are typically authorized through **aaa authorization commands**. However, the **no aaa authorization config-commands** command disables the authorization of configuration commands. In this state, authorization to execute configuration commands can be managed by controlling access to Global Configuration commands. The default setting authorizes configuration commands through the policy specified for all other commands.

- To enable the authorization of configuration commands with the policy specified for all other commands, enter **aaa authorization config-commands.**
- To require authorization of commands entered on the console, enter **aaa authorization console**.

By default, EOS does not verify authorization of commands entered on the console port.

Examples

- This command disables the authorization of configuration commands.

  ```
  switch(config)#no aaa authorization config-commands
  switch(config)#
  ```

- This command enables the authorization of configuration commands.

  ```
  switch(config)#aaa authorization config-commands
  switch(config)#
  ```

- This command configures the switch to authorize commands entered on the console, using the method specified through a previously executed **aaa authorization command**.

  ```
  switch(config)#aaa authorization console
  switch(config)#
  ```

4.3.4 Accounting

The accounting service collects information for billing, auditing, and reporting. The switch supports TACACS+ and RADIUS accounting by reporting user activity to either the TACACS+ server or RADIUS server in the form of accounting records. Accounting is enabled by the **aaa accounting** command.

The switch supports two types of accounting:

- EXEC: Provides information about user CLI sessions.
- Commands: Applies to the CLI commands a user issues. Command authorization attempts authorization for all commands, including configuration commands, associated with a specific privilege level.

4.4 Security Configuration Examples

These sections describe two sample TACACS+ host configuration.

4.4.1 Single Host Configuration

The single host configuration consists of a TACACS+ server with these attributes:

- IP address: 10.1.1.10
- encryption key: example_1
- port number: 49 (global default)
- timeout: 5 seconds (global default)

The switch authenticates the username and enable command against all TACACS+ servers which, in this case, is one host. If the TACACS+ server is unavailable, the switch authenticates with the local file.

Step 1 This step configures TACACS+ server settings – port number and timeout are global defaults.

```
switch(config)#tacacs-server host 10.1.1.10 key example_1
```

Step 2 This step configures the login authentication service.

```
switch(config)#aaa authentication login default group tacacs+ local
```

Step 3 This step configures the enable command password authentication service.

```
switch(config)#aaa authentication enable default group tacacs+ local
```

4.4.2 Multiple Host Configuration

The multiple host configuration consists of three TACACS+ servers at these locations:

- IP address 10.1.1.2 – port 49
- IP address 13.21.4.12 – port 4900
- IP address – 16.1.2.10 – port 49

The configuration combines the servers into these server groups:

- Bldg_1 group consists of the servers at 10.1.1.2 and 13.21.4.12
- Bldg_2 group consists of the servers at 16.1.2.10

All servers use these global TACACS+ defaults:

- encryption key – example_2
- timeout – 10 seconds

The switch authenticates these access methods:

- username access against Bldg_1 group then, if they are not available, against the local file.
- enable command against Bldg_2 group, then Bldg_1 group, then against the local file.

Step 1 TACACS+ Host commands:

These commands configure the IP address and ports for the three TACACS+ servers. The port for the first and third server is default 49.

```
switch(config)#tacacs-server host 10.1.1.12
switch(config)#tacacs-server host 13.21.4.12 port 4900
switch(config)#tacacs-server host 16.1.2.10
```

Step 2 Global Configuration Commands:

These commands configure the global encryption key and timeout values.

```
switch(config)#tacacs-server key example_2
switch(config)#tacacs-server timeout 10
```

Step 3 Group Server Commands:

The **aaa group server** commands create the server groups and place the CLI in server group configuration, during which the servers are placed in the group. The port number must be included if it is not the default port, as in the line that adds 13.21.4.12.

```
switch(config)#aaa group server tacacs+ Bldg_1
switch(config-sg-tacacs+-Bldg_1)#server 10.1.1.2
switch(config-sg-tacacs+-Bldg_1)#server 13.21.4.12 port 4900
switch(config-sg-tacacs+-Bldg_1)#exit
switch(config)#aaa group server tacacs+ Bldg_2
switch(config-sg-tacacs+-Bldg_2)#server 16.1.2.10
switch(config-sg-tacacs+-Bldg_2)#exit
switch(config)#
```

Step 4 Login and enable configuration authentication responsibility commands:

These commands configure the username and enable command password authentication services.

```
switch(config)#aaa authentication login default group Bldg_1 local
switch(config)#aaa authentication enable default group Bldg_1 group Bldg_2
local
```

4.5 AAA Commands

This section contains descriptions of the CLI commands that this chapter references.

Local Security File Commands

- username . . . . . Page 148
- username sshkey . . . . . Page 150
- aaa authentication policy local . . . . . Page 113
- enable secret . . . . . Page 124
- aaa root . . . . . Page 120
- show privilege . . . . . Page 139
- show user-account . . . . . Page 142
- show users . . . . . Page 143

Server (TACACS+ and RADIUS) Configuration Commands

- ip radius source-interface . . . . . Page 125
- ip tacacs source-interface . . . . . Page 126
- tacacs-server key . . . . . Page 146
- tacacs-server timeout . . . . . Page 147
- tacacs-server host . . . . . Page 144
- radius-server key . . . . . Page 130
- radius-server timeout . . . . . Page 132
- radius-server retransmit . . . . . Page 131
- radius-server deadtime . . . . . Page 127
- radius-server host . . . . . Page 128

Authentication, Authorization, and Accounting Commands

- aaa authentication login . . . . . Page 112
- aaa authentication enable . . . . . Page 111
- aaa authorization commands . . . . . Page 114
- aaa authorization exec . . . . . Page 117
- aaa authorization config-commands . . . . . Page 115
- aaa authorization console . . . . . Page 116
- aaa accounting . . . . . Page 110
- show aaa . . . . . Page 135

- clear aaa counters . . . . . Page 121
- clear aaa counters radius . . . . . Page 122
- clear aaa counters tacacs+ . . . . . Page 123

- show aaa counters . . . . . Page 136
- show aaa method-lists . . . . . Page 137
- show aaa sessions . . . . . Page 138

Server Group Configuration Commands

- aaa group server radius . . . . . Page 118
- aaa group server tacacs+ . . . . . Page 119
- server (server-group-RADIUS configuration mode) . . . . . Page 133
- server (server-group-TACACS+ configuration mode) . . . . . Page 134
- show radius . . . . . Page 140
- show tacacs . . . . . Page 141

aaa accounting

The **aaa accounting** command configures accounting method lists for a specified authorization type. Each list consists of a prioritized list of methods. The accounting module uses the first available listed method for the authorization type.

The **no aaa accounting** and **default aaa accounting** commands clear the specified method list by removing the corresponding **aaa accounting** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa accounting TYPE CONNECTION MODE [METHOD_1] [METHOD_2] ... [METHOD_N]
no aaa accounting TYPE CONNECTION
default aaa accounting TYPE CONNECTION
```

Parameters

- *TYPE* authorization type for which the command specifies a method list. Options include:
  - **EXEC** records user authentication events.
  - **COMMANDS ALL** records all entered commands.
  - **COMMANDS** *level* records entered commands of the specified *level* (ranges from 0 to 15).
- *CONNECTION* connection type of sessions for which method lists are reported. Options include:
  - **console** console connection.
  - **default** all connections not covered by other command options.
- *MODE* accounting mode that defines when accounting notices are sent. Options include:
  - **none** no notices are sent.
  - **start-stop** a *start* notice is sent when a process begins; a *stop* notice is sent when it ends.
  - **stop-only** a *stop* accounting record is generated after a process successfully completes.
- *METHOD_X* server groups (methods) to which the switch can send accounting records. The switch sends the method list to the first listed group that is available.

  Parameter value is not specified if ***MODE*** is set to ***none***. If ***MODE*** is not set to ***none,*** the command must provide at least one method. Each method is composed of one of the following:
  - **group** *name* the server group identified by *name.*
  - **group radius** server group that includes all defined RADIUS hosts.
  - **group tacacs+** server group that includes all defined TACACS+ hosts.

Examples

This command configures the switch to maintain start-stop accounting records for all command executed by switch users and submits them to all TACACS+ hosts.

```
switch(config)#aaa accounting commands all default start-stop group tacacs+
switch(config)#
```

This command configures the switch to maintain stop accounting records for all user EXEC sessions performed through the console and submits them to all TACACS+ hosts.

```
switch(config)#aaa accounting exec console stop group tacacs+
switch(config)#
```

aaa authentication enable

The **aaa authentication enable** command configures the service list that the switch references to authorize access to Privileged EXEC command mode.

The list consists of a prioritized list of service options. Available service options include:

- a named server group
- all defined TACACS+ hosts
- all defined RADIUS hosts
- local authentication
- no authentication

The switch authorizes access by using the first listed service option that is available.

When the list is not configured, it is set to **local**.

The **no aaa authentication enable** and **default aaa authentication enable** commands revert the list configuration as **local** by removing the **aaa authentication enable** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa authentication enable default METHOD_1 [METHOD_2] ... [METHOD_N]
no aaa authentication enable default
default aaa authentication enable default
```

Parameters

- *METHOD_X* authentication service method list. The command must provide at least one method. Each method is composed of one of the following:
  - **group** *name* the server group identified by *name*.
  - **group radius** a server group that consists of all defined RADIUS hosts.
  - **group tacacs+** a server group that consists of all defined TACACS+ hosts.
  - **local** local authentication.
  - **none** users are not authenticated; all access attempts succeed.

Example

This command configures the switch to authenticate the enable password through all configured TACACS+ servers. Local authentication is the backup if TACACS+ servers are unavailable.

```
switch(config)#aaa authentication default enable group TACACS+ local
switch(config)#
```

aaa authentication login

The **aaa authentication login** command configures service lists the switch references to authenticate usernames. Service lists consist of service options ordered by usage priority. The switch authenticates usernames through the first available service option. Supported service options include:

- a named server group
- all defined TACACS+ hosts
- all defined RADIUS hosts
- local authentication
- no authentication

The switch supports a ***console*** list for authenticating usernames through the console and a default list for authenticating usernames through all other connections.

- When the ***console*** list is not configured, the console connection uses the ***default*** list.
- When the ***default*** list is not configured, it is set to *local*.

The **no aaa authentication login** and **default aaa authentication login** commands revert the specified list configuration to its default by removing the corresponding **aaa authentication** login command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa authentication login CONNECTION SERVICE_1 [SERVICE_2] ... [SERVICE_N]
no aaa authentication login CONNECTION
default aaa authentication login CONNECTION
```

Parameters

- ***CONNECTION*** connection type of sessions for which authentication list is used
  - **default** the default authentication list.
  - **console** the authentication list for console logins.
- ***SERVICE_X*** an authentication service. Settings include:
  - **group** *name* identifies a previously defined server group.
  - **group radius** a server group that consists of all defined RADIUS hosts.
  - **group tacacs+** a server group that consists of all defined TACACS+ hosts.
  - **local** local authentication.
  - **none** the switch does not perform authentication. All access attempts succeed.

Examples

- This command configures the switch to authenticate usernames through the TAC-1 server group. The local database is the backup method if TAC-1 servers are unavailable.

```
switch(config)#aaa authentication login default group TAC-1 local
switch(config)#
```

- This command configures the switch to authenticate usernames through all TACACS+ servers, then all RADIUS servers if the TACACS+ servers are not available. If the RADIUS servers are also unavailable, the switch allows access to all login attempts without authentication.

```
switch(config)#aaa authenticatio login default group tacacs+ group radius none
switch(config)#
```

aaa authentication policy local

The **aaa authentication policy local allow-nopassword-remote-login** command permits usernames without passwords to log in from any port. The default switch setting only allows unprotected usernames to log in from the console.

The **no aaa authentication policy local allow-nopassword-remote-login** and **default aaa authentication policy local allow-nopassword-remote-login** commands return the switch to the default setting of denying unprotected usernames to log in except from the console.

Command Mode

Global Configuration

Command Syntax

```
aaa authentication policy local allow-nopassword-remote-login
no aaa authentication policy local allow-nopassword-remote-login
default aaa authentication policy local allow-nopassword-remote-login
```

Examples

- This command configures the switch to allow unprotected usernames to login from any port.

```
switch(config)#aaa authentication policy local allow-nopassword-remote-login
switch(config)#
```

- This command configures the switch to allow unprotected usernames to login only from the console port.

```
switch(config)#no aaa authenticatio policy local allow-nopassword-remote-login
switch(config)#
```

aaa authorization commands

The **aaa authorization commands** command configures the service list that authorizes CLI command access. All switch commands are assigned a privilege level that corresponds to the lowest level command mode from which it can be executed:

- Level 1: Commands accessible from EXEC mode.
- Level 15: Commands accessible from any mode except EXEC.

Command usage is authorized for each privilege level specified in the command.

The list consists of a prioritized list of service options. The switch authorizes access by using the first listed service option that is available. The available service options include:

- a named server group
- all defined TACACS+ hosts
- all defined RADIUS hosts
- local authorization
- no authorization

The list is set to **none** for all unconfigured privilege levels, allowing all CLI access attempts to succeed.

The **no aaa authorization commands** and **default aaa authorization commands** commands revert the list contents to ***none*** for the specified privilege levels.

Command Mode

Global Configuration

Command Syntax

```
aaa authorization commands PRIV default SERVICE_1 [SERVICE_2] ... [SERVICE_N]
no aaa authorization commands PRIV default
default aaa authorization commands PRIV default
```

Parameters

- ***PRIV*** Privilege levels of the commands. Options include:
  - — *level* numbers from **0** and **15.** Number, range, comma-delimited list of numbers and ranges.
  - — **all** commands of all levels.
- ***SERVICE_X*** Authorization service. Command must list at least one service. Options include:
  - — **group** *name* the server group identified by *name.*
  - — **group tacacs+** a server group that consists of all defined TACACS+ hosts.
  - — **local** local authorization.
  - — **none** the switch does not perform authorization. All access attempts succeed.

Examples

- This command authorizes configuration commands (privilege level 15) through the local file. The switch denies command access to users not listed in the local file.

```
switch(config)#aaa authorization commands 15 default local
switch(config)#
```

- This command authorizes all commands entered on the CLI.

```
switch(config)#aaa authorization commands all default none
switch(config)#
```

aaa authorization config-commands

The **aaa authorization config-commands** command enables authorization of commands in any configuration mode, such as Global Configuration and Interface Configuration modes. Commands are authorized through the policy specified by the **aaa authorization commands** setting. This command is enabled by default and does not appear in *running-config*. Issuing this command has no effect unless running-config contains the **no aaa authorization config-commands** command.

The **no aaa authorization config-commands** command disables configuration command authorization. When configuration command authorization is disabled, *running-config* contains the **no aaa authorization config-commands** command. The **default aaa authorization config-commands** command restores the default setting by removing the **no aaa authorization config-commands** from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
aaa authorization config-commands
no aaa authorization config-commands
default aaa authorization config-commands
```

Example

- This command disables the authorization of configuration commands.

```
switch(config)#no aaa authorization config-commands
switch(config)#
```

- This command enables the authorization of configuration commands.

```
switch(config)#aaa authorization config-commands
switch(config)#
```

aaa authorization console

The **aaa authorization console** command configures the switch to authorize commands entered through the console. By default, commands entered through the console do not require authorization.

The **no aaa authorization console** and **default aaa authorization console** commands restore the default setting.

Command Mode

Global Configuration

Command Syntax

```
aaa authorization console
no aaa authorization console
default aaa authorization console
```

Example

- This command configures the switch to authorize commands entered on the console, using the method specified through an previously executed **aaa authorization command**.

```
switch(config)#aaa authorization console
switch(config)#
```

aaa authorization exec

The **aaa authorization exec** command configures the service list that the switch references to authorize access to open an EOS CLI shell.

The list consists of a prioritized list of service options. The switch authorizes access by using the first listed service option to which the switch can connect. When the switch cannot communicate with an entity that provides a specified service option, it attempts to use the next option in the list.

The available service options include:

- a named server group
- all defined TACACS+ hosts
- all defined RADIUS hosts
- local authentication
- no authentication

When the list is not configured, it is set to **none**, allowing all CLI access attempts to succeed.

The **no aaa authorization exec** and **default aaa authorization exec** commands set the list contents to **none**.

Command Mode

Global Configuration

Command Syntax

```
aaa authorization exec default METHOD_1 [METHOD_2] ... [METHOD_N]
no aaa authorization exec default
default aaa authorization exec default
```

Parameters

- *METHOD_X* authorization service (method). The switch uses the first listed available method.

  The command must provide at least one method. Each method is composed of one of the following:

  — **group** *name* the server group identified by *name*.
  — **group radius** a server group that consists of all defined RADIUS hosts.
  — **group tacacs+** a server group that consists of all defined TACACS+ hosts.
  — **local** local authentication.
  — **none** the switch does not perform authorization. All access attempts succeed.

Guidelines

During the exec authorization process, the TACACS+ server response may include attribute-value (AV) pairs. The switch recognizes **priv-lvl=x** (where **x** is an integer between 0 and 15), which is a mandatory AV pair. A TACACS+ server that sends any other mandatory AV pair is denied access to the switch. The receipt of optional AV pairs by the switch has no affect on decisions to permit or deny access to the TACACS+ server.

Example

- This command specifies that the TACACS+ servers authorize users that attempt to open an EOS CLI shell.

```
switch(config)#aaa authorization exec default group tacacs+
switch(config)#
```

aaa group server radius

The **aaa group server radius** command enters the server-group-radius configuration mode for the specified group name. The command creates the specified group if it was not previously created. Commands are available to add servers to the group.

A server group is a collection of servers that are associated with a single label. Subsequent authorization and authentication commands access all servers in a group by invoking the group name. Server group members must be previously configured with a **radius-server host** command.

The **no aaa group server radius** and **default aaa group server radius** commands delete the specified server group from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa group server radius group_name
no aaa group server radius group_name
default aaa group server radius group_name
```

Parameters

- *group_name* name (text string) assigned to the group. Cannot be identical to a name already assigned to a TACACS+ server group.

Commands Available in server-group-radius Configuration Mode

- **server (server-group-RADIUS configuration mode)**

Related Commands

- **aaa group server tacacs+**

Example

- This command creates the RADIUS server group named RAD-SV1 and enters server group configuration mode for the new group.

```
switch(config)#aaa group server radius RAD-SV1
switch(config-sg-radius-RAD-SV1)#
```

aaa group server tacacs+

The **aaa group server tacacs+** command enters server-group-tacacs+ configuration mode for the specified group name. The command creates the specified group if it was not previously created. Commands are available to add servers to the group.

A server group is a collection of servers that are associated with a single label. Subsequent authorization and authentication commands access all servers in a group by invoking the group name. Server group members must be previously configured with a **tacacs-server host** command.

The **no aaa group server tacacs+** and **default aaa group server tacacs+** commands delete the specified server group from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
aaa group server tacacs+ group_name
no aaa group server tacacs+ group_name
default aaa group server tacacs+ group_name
```

Parameters

- *group_name* name (text string) assigned to the group. Cannot be identical to a name already assigned to a RADIUS server group.

Commands Available in server-group-tacacs+ Configuration Mode

- **server (server-group-TACACS+ configuration mode)**

Related Commands

- **aaa group server radius**

Example

- This command creates the TACACS+ server group named TAC-GR and enters server group configuration mode for the new group.

```
switch(config)#aaa group server tacacs+ TAC-GR
switch(config-sg-tacacs+-TAC-GR)#
```

aaa root

The **aaa root** command specifies the password security level for the root account and can assign a password to the account.

The **no aaa root** and **default aaa root** commands disable the root account by removing the **aaa root** command from ***running-config***. The root account is disabled by default.

Command Mode

Global Configuration

Command Syntax

```
aaa root SECURITY_LEVEL [ENCRYPT_TYPE] [password]
no aaa root
default aaa root
```

Parameters

- ***SECURITY_LEVEL*** password assignment level. Settings include
  - **secret** the root account is assigned to the password.
  - **nopassword** the root account is not password protected.
- ***ENCRYPT_TYPE*** encryption level of the *password* parameter. This parameter is present only when ***SECURITY_LEVEL*** is **secret**. Settings include:
  - <no parameter> the password is entered as clear text.
  - **0** the password is entered as clear text. Equivalent to <no parameter>.
  - **5** the password is entered as an md5 encrypted string.
- *password* text that authenticates the username. The command includes this parameter only if ***SECURITY_LEVEL*** is **secret**.
  - *password* must be in clear text if ***ENCRYPT_TYPE*** specifies clear text.
  - *password* must be an encrypted string if ***ENCRYPT_TYPE*** specifies an encrypted string.

  Encrypted strings entered through this parameter are generated elsewhere.

Examples

- These equivalent commands assign **f4980** as the root account password.

```
switch(config)#aaa root secret f4980

switch(config)#aaa root secret 0 f4980
```

- This command assigns the text (ab234) that corresponds to the encrypted string of 1HW05LEY8$QEVw6JqjD9VqDfh.O8r.b. as the root password.

```
switch(config)#aaa root secret 5 $1$HW05LEY8$QEVw6JqjD9VqDfh.O8r.b
switch(config)#
```

- This command removes the password from the root account.

```
switch(config)#aaa root nopassword
switch(config)#
```

- This command disables the root login.

```
switch(config)#no aaa root
switch(config)#
```

clear aaa counters

The **clear aaa counters** command resets the counters that track the number of service transactions performed by the switch since the last time the counters were reset. The **show aaa counters** command displays the counters reset by the **clear aaa counters** command.

Command Mode

Privileged EXEC

Command Syntax

clear aaa counters [*SERVICE_TYPE*]

Example

- These commands display the effect of the **clear aaa counters** command on the aaa counters.

```
switch#clear aaa counters
switch#show aaa counters
Authentication
          Successful:          0
              Failed:          0
 Service unavailable:          0

Authorization
             Allowed:          1
              Denied:          0
 Service unavailable:          0

Accounting
          Successful:          0
               Error:          0
             Pending:          0

Last time counters were cleared: 0:00:44 ago
```

clear aaa counters radius

The **clear aaa counters radius** command resets the counters that track the statistics for the RADIUS servers that the switch access. The **show radius** command displays the counters reset by the **clear aaa counters radius** command.

Command Mode

Privileged EXEC

Command Syntax

```
clear aaa counters radius
```

Example

- These commands display the effect of the **clear aaa counters radius** command on the RADIUS counters.

```
switch#show radius
RADIUS server            : radius/10
          Connection opens:        204
         Connection closes:          0
    Connection disconnects:        199
       Connection failures:         10
       Connection timeouts:          2
             Messages sent:       1490
         Messages received:       1490
            Receive errors:          0
          Receive timeouts:          0
             Send timeouts:          0

Last time counters were cleared: never
switch#clear aaa counters radius
switch#show radius
RADIUS server            : radius/10
          Connection opens:          0
         Connection closes:          0
    Connection disconnects:          0
       Connection failures:          0
       Connection timeouts:          0
             Messages sent:          0
         Messages received:          0
            Receive errors:          0
          Receive timeouts:          0
             Send timeouts:          0

Last time counters were cleared: 0:00:03 ago
switch#
```

clear aaa counters tacacs+

The **clear aaa counters tacacs+** command resets the counters that track the statistics for the TACACS+ servers that the switch access. The **show tacacs** command displays the counters reset by the **clear aaa counters tacacs+** command.

Command Mode

Privileged EXEC

Command Syntax

```
clear aaa counters tacacs+
```

Example

- These commands display the effect of the **clear aaa counters tacacs+** command on the tacacs+ counters.

```
switch#show tacacs
TACACS+ server            : tacacs/49
          Connection opens:      15942
         Connection closes:          7
    Connection disconnects:       1362
       Connection failures:          0
       Connection timeouts:          0
             Messages sent:      34395
         Messages received:      34392
            Receive errors:          0
           Receive timeouts:          2
              Send timeouts:          0

Last time counters were cleared: never

TACACS+ source-interface: Enabled
  TACACS+ outgoing packets will be sourced with an IP address associated with
the Loopback0 interface
switch#clear aaa counters tacacs+
switch#show tacacs
TACACS+ server            : tacacs/49
          Connection opens:          0
         Connection closes:          0
    Connection disconnects:          0
       Connection failures:          0
       Connection timeouts:          0
             Messages sent:          0
         Messages received:          0
            Receive errors:          0
           Receive timeouts:          0
              Send timeouts:          0

Last time counters were cleared: 0:00:03 ago
switch#

TACACS+ source-interface: Enabled
  TACACS+ outgoing packets will be sourced with an IP address associated with
the Loopback0 interface
switch#
```

enable secret

The **enable secret** command creates a new enable password or changes an existing password.

The **no enable secret** and **default enable secret** commands delete the **enable password** by removing the **enable secret** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
enable secret [ENCRYPT_TYPE] password
no enable secret
default enable secret
```

Parameters

- ***ENCRYPT_TYPE*** encryption level of the *password* parameter. Settings include:
  - <no parameter> the password is entered as clear text.
  - **0** the password is entered as clear text. Equivalent to <no parameter>.
  - **5** the password is entered as an md5 encrypted string.
- *password* text that authenticates the username.
  - *password* must be in clear text if ***ENCRYPT_TYPE*** specifies clear text.
  - *password* must be an encrypted string if ***ENCRYPT_TYPE*** specifies an encrypted string.

  Encrypted strings entered through this parameter are generated elsewhere.

Examples

- These equivalent commands assign *xyrt1* as the enable password.

```
switch(config)#enable secret xyrt1

switch(config)#enable secret 0 xyrt1
```

- This command assigns the enable password to the clear text (12345) that corresponds to the encrypted string *$1$8bPBrJnd$Z8wbKLHpJEd7d4tc5Z/6h/*. The string was generated by an MD5-encryption program using 12345 as the seed.

```
switch(config)#enable secret 5 $1$8bPBrJnd$Z8wbKLHpJEd7d4tc5Z/6h/
switch(config)#
```

- This command deletes the enable password.

```
switch(config)#no enable secret
switch(config)#
```

ip radius source-interface

The **ip radius source-interface** command specifies the interface from which the IPv4 address is derived for use as the source for outbound RADIUS packets. When a source interface is not specified, the switch selects an interface.

The **no ip radius source-interface** and **default ip radius source-interface** commands remove the **ip radius source-interface** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip radius source-interface INT_NAME
no ip radius source-interface
default ip radius source-interface
```

Parameters

- ***INT_NAME*** Interface type and number. Options include:
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface management** *m_num* Management interface specified by *m_num*.
  - **interface port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command configures the source address for outbound RADIUS packets as the IPv4 address assigned to the loopback interface.

```
switch(config)#ip radius source-interface loopback 0
switch(config)#
```

ip tacacs source-interface

The **ip tacacs source-interface** command specifies the interface from which the IPv4 address is derived for use as the source for outbound TACACS+ packets. When a source interface is not specified, the switch selects an interface.

The **no ip tacacs source-interface** and **default ip tacacs source-interface** commands remove the **ip tacacs source-interface** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip tacacs source-interface INT_NAME
no ip tacacs source-interface
default ip tacacs source-interface
```

Parameters

- ***INT_NAME*** Interface type and number. Options include:
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface management** *m_num* Management interface specified by *m_num*.
  - **interface port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.

Guidelines

Configuring a TACACS+ source interface for user defined VRFs is not supported. The **show vrf** command displays VRF interface configuration.

Example

- This command configures the source address for outbound TACACS+ packets as the IPv4 address assigned to the loopback interface.

```
switch(config)#ip tacacs source-interface loopback 0
switch(config)#
```

radius-server deadtime

The **radius-server deadtime** command defines global deadtime period, when the switch ignores a non-responsive RADIUS server. A non-responsive server is one that failed to answer any attempt to retransmit after a timeout expiry. Deadtime is disabled if a value is not configured.

The **no radius-server deadtime** and **default radius-server deadtime** commands restore the default global deadtime period of three minutes by removing the **radius-server deadtime** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
radius-server deadtime dead_interval
no radius-server deadtime
default radius-server deadtime
```

Parameters

- *dead_interval* period that the switch ignores non-responsive servers (minutes). Value ranges from 1 to 1000. Default is 3.

Related Commands

- **radius-server host**

Example

- This command programs the switch to ignore a server for two hours if it fails to respond to a request during the period defined by timeout and retransmit parameters.

```
switch(config)#radius-server deadtime 120
switch(config)#
```

radius-server host

The **radius-server host** command sets parameters for communicating with a specific RADIUS server. These values override global settings when the switch communicates with the specified server.

A RADIUS server is defined by its server address, authorization port, and accounting port. Servers with different address-authorization port-accounting port combinations have separate configurations.

The **no radius-server host** and **default radius-server** commands remove settings for the RADIUS server configuration at the specified address-authorization port-accounting port location by deleting the corresponding **radius-server host** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
radius-server host ADDR [AUTH] [ACCT] [TIMEOUT] [DEAD] [RETRAN] [ENCRYPT_KEY]
no radius-server host [ADDR] [AUTH] [ACCT]
default radius-server host [ADDR] [AUTH] [ACCT]
```

Parameters

- ***ADDR*** RADIUS server location. Options include:
  - *ipv4_addr* server's IPv4 address.
  - *host_name* server's DNS host name (FQDN).
- ***AUTH*** Authorization port number.
  - <no parameter> default port of 1812.
  - **auth-port** *number* *number* ranges from 1 to 65535.
- ***ACCT*** Accounting port number.
  - <no parameter> default port of 1813.
  - **acct-port** *number* *number* ranges from 1 to 65535.
- ***TIMEOUT*** timeout period (seconds). Ranges from 1 to 1000.
  - <no parameter> assigns global timeout value (see **radius-server timeout**).
  - **timeout** *number* assigns *number* as the timeout period. Ranges from 1 to 1000.
- ***DEAD*** period (minutes) when the switch ignores a non-responsive RADIUS server.
  - <no parameter> assigns global deadtime value (see **radius-server deadtime**).
  - **deadtime** *number* specifies deadtime, where *number* ranges from 1 to 1000.
- ***RETRAN*** attempts to access RADIUS server after the first timeout expiry.
  - <no parameter> assigns global retransmit value (see **radius-server retransmit**).
  - **retransmit** *number* specifies number of attempts, where *number* ranges from 1 to 100.
- ***ENCRYPT_KEY*** encryption key that switch and server use to communicate.
  - <no parameter> assigns global encryption key (see **radius-server key**).
  - **key** *key_text* where *key_text* is in clear text.
  - **key 5** *key_text* where *key_text* is in clear text.
  - **key 7** *key_text* where *key_text* is provide in an encrypted string.

Examples

- This command configures the switch to communicate with the RADIUS server located at *10.1.1.5*. The switch uses the global timeout, deadtime, retransmit, and key settings to communicate with this server, and communicates through port 1812 for authorization and 1813 for accounting.

```
switch(config)#radius-server host 10.1.1.5
switch(config)#
```

- This command configures the switch to communicate with the RADIUS server assigned the host name *RAD-1*. Communication for authorization is through port 1850; communication for accounting is through port 1813 (the default).

```
switch(config)#radius-server host RAD-1 auth-port 1850
switch(config)#
```

radius-server key

The **radius-server key** command defines the global encryption key the switch uses when communicating with any RADIUS server for which a key is not defined.

The **no radius-server key** and **default radius-server key** commands remove the global key from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
radius-server key [ENCRYPT_TYPE] encrypt_key
no radius-server key
default radius-server key
```

Parameters

- ***ENCRYPT_TYPE*** encryption level of *encrypt_key*.
  - — <no parameter> encryption key is entered as clear text.
  - — **0** encryption key is entered as clear text. Equivalent to <no parameter>.
  - — **7** *encrypt_key* is an encrypted string.
- *encrypt_key* shared key that authenticates the username.
  - — *encrypt_key* must be in clear text if ***ENCRYPT_TYPE*** specifies clear text.
  - — *encrypt_key* must be an encrypted string if ***ENCRYPT_TYPE*** specifies an encrypted string.

  Encrypted strings entered through this parameter are generated elsewhere.

Related Commands

- **radius-server host**

Examples

- This command configures *cv90jr1* as the global encryption key.

```
switch(config)#radius-server key 0 cv90jr1
switch(config)#
```

- This command assigns *cv90jr1* as the key by specifying the corresponding encrypted string.

```
switch(config)#radius-server key 7 020512025B0C1D70
switch(config)#
```

radius-server retransmit

The **radius-server retransmit** command defines the global retransmit count, which specifies the number of times the switch attempts to access the RADIUS server after the first timeout expiry.

The **no radius-server retransmit** and **default radius-server retransmit** commands restore the global retransmit count to its default value of three by deleting the **radius-server retransmit** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
radius-server retransmit count
no radius-server retransmit
default radius-server retransmit
```

Parameters

- *count* retransmit attempts after first timeout expiry. Settings range from 1 to 100. Default is 3.

Related Commands

- **radius-server host**

Example

- This command configures the switch to attempt five RADIUS server contacts after the initial timeout. If the timeout parameter is set to 50 seconds, then the total period that the switch waits for a response is ((5+1)*50) = 300 seconds.

```
switch(config)#radius-server retransmit 5
switch(config)#
```

radius-server timeout

The **radius-server timeout** command defines the global timeout the switch uses when communicating with any RADIUS server for which a timeout is not defined.

The **no radius-server timeout** and **default radius-server timeout** commands restore the global timeout default period of five seconds by removing the **radius-server timeout** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
radius-server timeout time_period
no radius-server timeout
default radius-server timeout
```

Parameters

- *time_period* timeout period (seconds). Range from 1 to 1000. Default is 5.

Related Commands

- **radius-server host**

Example

- This command configures the switch to wait 50 seconds for a RADIUS server response before issuing an error.

```
switch(config)#radius-server timeout 50
switch(config)#
```

server (server-group-RADIUS configuration mode)

The **server (server-group-RADIUS configuration mode)** command adds the specified RADIUS server to the configuration mode group. Servers must be configured with the **radius-server host** command before adding them to the server group.

A RADIUS server is defined by its server address, authorization port, and accounting port. A group can contain multiple servers with the same IP address that have different authorization or accounting ports.

The **no server** and **default server** commands remove the specified server from the group.

Command Mode

Server-Group-RADIUS Configuration

Command Syntax

```
server ADDR [AUTH] [ACCT]
no server ADDR [AUTH] [ACCT]
default server ADDR [AUTH] [ACCT]
```

Parameters

- ***ADDR*** RADIUS server location. Options include:
  - *ipv4_addr* server's IPv4 address.
  - *host_name* server's DNS host name (FQDN).
- ***AUTH*** Authorization port number.
  - <no parameter> default port of 1812.
  - **auth-port** *number* *number* ranges from 1 to 65535.
- ***ACCT*** Accounting port number.
  - <no parameter> default port of 1813.
  - **acct-port** *number* *number* ranges from 1 to 65535.

Example

- These commands add two servers to the RAD-SV1 server group.

```
switch(config)#aaa group server radius RAD-SV1
switch(config-sg-radius-RAD-SV1)#server RAC-1
switch(config-sg-radius-RAD-SV1)#server 10.1.5.14 acct-port 1851
switch(config-sg-radius-RAD-SV1)#
```

server (server-group-TACACS+ configuration mode)

The **server (server-group-TACACS+ configuration mode)** command adds the specified TACACS+ server to the configuration mode group. Servers must be configured with the **tacacs-server host** command before adding them to the server group.

A TACACS+ server is defined by its server address and port number. Servers with different address-port combinations have separate statements in ***running-config***.

The **no server** and **default server** commands remove the specified server from the group.

Command Mode

Server-Group-TACACS+ Configuration

Command Syntax

```
server LOCATION [PORT]
no server LOCATION [PORT]
default server LOCATION [PORT]
```

Parameters

- ***LOCATION*** TACACS+ server location. Options include:
  - *ipv4_addr* server's IPv4 address.
  - *ipv6_addr* server's IPv6 address.
  - *host_name* server's DNS host name (FQDN).
- ***PORT*** TCP connection port number.
  - <no parameter> default port of 49.
  - **port** *number* *number* ranges from 1 to 65535.

Example

- These commands add two servers to the TAC-GR server group with default port number 49.

```
switch(config)#aaa group server tacacs+ TAC-GR
switch(config-sg-tacacs+-TAC-GR)#server TAC-1
switch(config-sg-tacacs+-TAC-GR)#server 10.1.4.14
switch(config-sg-tacacs+-TAC-GR)#
```

show aaa

The **show aaa** command displays the user database. The command displays the encrypted enable password first, followed by a table of usernames and their corresponding encrypted password.

The command does not display unencrypted passwords.

Command Mode

Privileged EXEC

Command Syntax

show aaa

Example

- This command configures the switch to authenticate the enable password through all configured TACACS+ servers. Local authentication is the backup if TACACS+ servers are unavailable.

```
switch#show aaa
Enable password (encrypted): $1$UL4gDWy6$3KqCPYPGRvxDxUq3qA/Hs/
Username  Encrypted passwd
--------  --------------------------------
admin
janis     $1$VVnDH/Ea$iwsfnrGNO8nbDsf0tazp9/
thomas    $1$/MmXTUil$.fJxLfcumzppNSEDVDWq9.
switch#
```

show aaa counters

The **show aaa counters** command displays the number of service transactions performed by the switch since the last time the counters were reset.

Command Mode

Privileged EXEC

Command Syntax

show aaa counters

Example

- This command displays the number of authentication, authorization, and accounting transactions.

```
switch#show aaa counters
Authentication
          Successful:          30
              Failed:           0
 Service unavailable:           0

Authorization
             Allowed:         188
              Denied:           0
 Service unavailable:           0

Accounting
          Successful:           0
               Error:           0
             Pending:           0

Last time counters were cleared: never
switch#
```

show aaa method-lists

The **show aaa method-lists** command displays all the named method lists defined in the specified authentication, authorization, and accounting (AAA) service.

Command Mode

Privileged EXEC

Command Syntax

show aaa method-lists *SERVICE_TYPE*

Parameters

- *SERVICE_TYPE* the service type of the method lists that the command displays.
  - **accounting** accounting services.
  - **authentication** authentication services.
  - **authorization** authorization services.
  - **all** accounting, authentication, and authorization services.

Example

- This command configures the named method lists for all AAA services.

```
switch#show aaa method-lists all
Authentication method lists for LOGIN:
  name=default methods=group tacacs+, local
Authentication method list for ENABLE:
  name=default methods=local
Authorization method lists for COMMANDS:
  name=privilege0-15 methods=group tacacs+, local
Authentication method list for EXEC:
  name=exec methods=group tacacs+, local
Accounting method lists for COMMANDS:
  name=privilege0-15 default-action=none
Accounting method list for EXEC:
  name=exec default-action=none
switch#
```

show aaa sessions

The **show aaa sessions** command displays information about active AAA login sessions. Information includes username, TTY, state of the session (pending or established), duration, authentication method, and if available, remote host and remote username.

Command Mode

Privileged EXEC

Command Syntax

show aaa sessions

Example

- This command configures the switch to authenticate the enable password through all configured TACACS+ servers. Local authentication is the backup if TACACS+ servers are unavailable.

```
switch#show aaa sessions
Session  Username TTY        State Duration Auth Method  Rem. Host     Rem. User
-------- -------- ---------- ----- -------- ------------ ------------- ---------
306      admin    ssh        P     192:12:48 group tacacs+ local158.sm.comp.com
519      admin    ssh        E     95:54:28 group tacacs+ bs1.pa.comp.com
683      admin    ssh        E     21:54:45 group tacacs+ bs1.pa.comp.com
737      admin    ssh        E     00:19:49 group tacacs+ 172.22.6.104
switch#
```

show privilege

The **show privilege** command displays privilege level of the current CLI session.

Command Mode

EXEC

Command Syntax

```
show privilege
```

Example

- This command displays the current privilege level.

```
switch>show privilege
Current privilege level is 15
switch>
```

show radius

The **show radius** command displays statistics for the RADIUS servers that the switch accesses.

Command Mode

EXEC

Command Syntax

show radius

Example

- This command displays statistics for connected TACACS+ servers.

```
switch>show radius
RADIUS server            : radius/10
          Connection opens:        204
         Connection closes:          0
    Connection disconnects:        199
       Connection failures:         10
       Connection timeouts:          2
             Messages sent:       1490
         Messages received:       1490
            Receive errors:          0
          Receive timeouts:          0
             Send timeouts:          0

Last time counters were cleared: never
switch>
```

show tacacs

The **show tacacs** command displays statistics for the TACACS+ servers that the switch accesses.

Command Mode

EXEC

Command Syntax

show tacacs

Example

- This command displays statistics for connected TACACS+ servers.

```
switch>show tacacs
TACACS+ server            : tacacs/49
          Connection opens:      15942
         Connection closes:          7
    Connection disconnects:       1362
       Connection failures:          0
       Connection timeouts:          0
             Messages sent:      34395
         Messages received:      34392
            Receive errors:          0
          Receive timeouts:          2
             Send timeouts:          0

Last time counters were cleared: never

TACACS+ source-interface: Enabled
  TACACS+ outgoing packets will be sourced with an IP address associated with
the Loopback0 interface
switch>
```

show user-account

The **show user-account** command displays the names and privilege levels of users that are listed in *running-config*. The ssh public-key is also listed for names for which an SSH key is configured.

Command Mode
Privileged EXEC

Command Syntax
show user-account

Example

- This command displays the usernames that are configured on the switch.

```
switch#show user-account
user: admin
       privilege level: 1
user: FRED
       privilege level: 1
       ssh public key: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDjUg2VDiBX7In0q
HtN5PyHOWtYvIoeZsxF5YmesQ/rh++mbpT504dL7So+Bpr9T/0qIj+zilat8fX/JlO42+3pjfkHY
/+lsT2EPNjGTK7uJv1wSGmhc3+90dNmJtr5YVlJFjjQ5m+5Pa+PGe3z4JIV1lY2NhLrV2fXtbciL
djnj6FAlhXjiLt51DJhG13uUxGBJe0+NlGvpEsTJVJvMdJuS6weMi+xSXc9yQimVD2weJBHsYFng
hST2j0pAyF2S7/EOU13pY42RztDSs42nMNNrutPT0q5Z17aAKvhpd0dDlc+qIwrCrXbeIChHem7+
0N8/zA3alBK4eKSFSZBd3Pb admin@switch
switch#
```

show users

The **show users** command displays the usernames that are currently logged into the switch.

Command Mode
Privileged EXEC

Command Syntax
show users

Example

- This command displays the users that are logged into the switch.

```
switch#show users
    Line       User       Host(s)              Idle       Location
   1 vty 2     john       idle                    1d  10.22.6.113
   2 vty 4     jane       idle                21:33:00  10.22.26.26
*  3 vty 6     ted        idle                00:00:01  10.17.18.71

switch#
```

tacacs-server host

The **tacacs-server host** command sets communication parameters for communicating with a specific TACACS+ server. These values override global settings when the switch communicates with the specified server.

A TACACS+ server is defined by its server address and port number. Servers with different combinations of address-port-VRF-multiplex settings have separate statements in ***running-config***.

The **no tacacs-server host** and **default tacacs-server host** commands remove settings for the TACACS+ server configuration at the specified address-port-VRF combination by deleting the corresponding **tacacs-server host** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
tacacs-server host SERVER_ADDR [MULTIPLEX] [VRF_INST] [PORT] [TIMEOUT] [ENCRYPT]
no tacacs-server host [SERVER_ADDR] [MULTIPLEX] [VRF_INST] [PORT]
default tacacs-server host [SERVER_ADDR] [MULTIPLEX] [VRF_INST] [PORT]
```

Parameters

- ***SERVER_ADDR*** TACACS+ server location. Options include:
  - *ipv4_addr* server's IPv4 address.
  - *ipv6_addr* server's IPv6 address.
  - *host_name* server's DNS host name (FQDN).
- ***MULTIPLEX*** TACACS+ server support of multiplex sessions on a TCP connection.
  - <no parameter> server does not support multiplexing.
  - **single-connection** server supports session multiplexing.
- ***VRF_INST*** specifies the VRF instance used to communicate with the specified server.
  - <no parameter> switch communicates with the server using the main VRF.
  - **vrf** *vrf_name* switch communicates with the server using the specified user-defined VRF.
- ***PORT*** port number of the TCP connection.
  - <no parameter> default port of 49.
  - **port** *number* port *number* ranges from 1 to 65535.
- ***TIMEOUT*** timeout period (seconds).
  - <no parameter> assigns the globally configured timeout value (see **tacacs-server timeout**).
  - **timeout** *number* timeout period (seconds). *number* ranges from 1 to 1000.
- ***ENCRYPT*** encryption key the switch and server use to communicate. Settings include
  - <no parameter> assigns the globally configured encryption key (see **tacacs-server key**).
  - **key** *key_text* where *key_text* is in clear text.
  - **key 5** *key_text* where *key_text* is in clear text.
  - **key 7** *key_text* where *key_text* is an encrypted string.

Examples

- This command configures the switch to communicate with the TACACS+ server located at *10.1.1.5*. The switch uses the global timeout, encryption key, and port settings.

```
switch(config)#tacacs-server host 10.1.1.5
switch(config)#
```

- This command configures the switch to communicate with the TACACS+ server assigned the host name *TAC_1*. The switch defines the timeout period as *20 seconds* and the encryption key as *rp31E2v*.

```
switch(config)#tacacs-server host TAC_1 timeout 20 key rp31E2v
switch(config)#
```

- This command configures the switch to communicate with the TACACS+ server located at *10.12.7.9,* indicates that the server supports multiplexing sessions on the same TCP connection, and that access is through port 54.

```
switch(config)#tacacs-server host 10.12.7.9 single-connection port 54
switch(config)#
```

tacacs-server key

The **tacacs-server key** command defines the global encryption key the switch uses when communicating with any TACACS+ server for which a key is not defined.

The **no tacacs-server key** and **default tacacs-server key** commands remove the global key from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
tacacs-server key [ENCRYPT_TYPE] encrypt_key
no tacacs-server key
default tacacs-server key
```

Parameters

- ***ENCRYPT_TYPE*** encryption level of *encrypt_key*.
  - — <no parameter> encryption key is entered as clear text.
  - — **0** encryption key is entered as clear text. Equivalent to <no parameter>.
  - — **7** *encrypt_key* is an encrypted string.
- *encrypt_key* shared key that authenticates the username.
  - — *encrypt_key* must be in clear text if ***ENCRYPT_TYPE*** specifies clear text.
  - — *encrypt_key* must be an encrypted string if ***ENCRYPT_TYPE*** specifies an encrypted string.

  Encrypted strings entered through this parameter are generated elsewhere.

Related Commands

- **tacacs-server host**

Examples

- This command configures *cv90jr1* as the encryption key.

```
switch(config)#tacacs-server key 0 cv90jr1
switch(config)#
```

- This command assigns *cv90jr1* as the key by specifying the corresponding encrypted string.

```
switch(config)#tacacs-server key 7 020512025B0C1D70
switch(config)#
```

tacacs-server timeout

The **tacacs-server timeout** command defines the global timeout the switch uses when communicating with any TACACS+ server for which a timeout is not defined.

The **no tacacs-server timeout** and **default tacacs-server timeout** commands restore the global timeout default period of five seconds by removing the **tacacs-server timeout** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
tacacs-server timeout time_period
no tacacs-server timeout
default tacacs-server timeout
```

Parameters

- *time_period* timeout period (seconds). Settings range from 1 to 1000. Default is 5.

Related Commands

- **tacacs-server host**

Example

- This command configures the switch to wait 20 seconds for a TACACS+ server response before issuing an error.

```
switch(config)#tacacs-server timeout 20
switch(config)#
```

username

The **username** command adds a username to the local file and assigns a password to a username. If the command specifies an existing username, the command replaces the password in the local file. The command can define a username without a password or remove the password from a username.

The **no username** and **default username** commands delete the specified username by removing the corresponding **username** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
username name [PRIVILEGE_LEVEL] SECURITY [ENCRYPTION] [password]
no username name
default username name
```

Parameters

- *name* username text that the user enters at the login prompt to access the CLI.

  Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

  ```
  @   #   $   %   ^   &   *   (   )   -   _   =
  +   {   }   [   ]   ;   <   >   ,   .   ~   |
  ```

- ***PRIVILEGE_LEVEL*** user's initial session privilege level. This parameter is used when an authorization command includes the local option.

  — <no parameter> the privilege level is set to 1.
  — **privilege** *rank* where *rank* is an integer between 0 and 15.

- ***SECURITY*** password assignment option.

  — **secret** username is assigned to the specified password.
  — **nopassword** username is not password protected.

- ***ENCRYPTION*** encryption level of the password. Included only if ***SECURITY*** is **secret**.

  — <no parameter> *password* is a clear text string.
  — **0** the *password* is a clear text string. Equivalent to the <no parameter> case.
  — **5** the *password* is an md5 encrypted string.

- *password* text that authenticates the username. Included only if ***SECURITY*** is **secret**.

  — *password* is a clear text string if ***ENCRYPTION*** specifies clear text
  — *password* is an encrypted string if ***ENCRYPTION*** specifies an encrypted string.

  Encrypted strings entered through this parameter are generated elsewhere. The encryption option is typically used to enter a list of username-passwords from a script.

Examples

- These equivalent commands create the username *john* and assigns it the password *x245*. The password is entered in clear text because the ***ENCRYPTION*** parameter is either omitted or zero.

  ```
  switch(config)#username john secret x245

  switch(config)#username john secret 0 x245
  ```

- This command creates the username *john* and assigns it to the text password that corresponds to the encrypted string *1sU.7hptc$TsJ1qslCL7ZYVbyXNG1wg1.* The string was generated by an MD5-encryption program using x245 as the seed.

```
switch(config)#username john secret 5 $1$sU.7hptc$TsJ1qslCL7ZYVbyXNG1wg1
switch(config)#
```

  A user authenticates the username *john* by entering ***x245*** when the CLI prompts for a password.

- This command creates the username *jane* without securing it with a password. It also removes a password if the *jane* username exists.

```
switch(config)#username jane nopassword
switch(config)#
```

- This command removes the username **william** from the local file.

```
switch(config)#no username william
switch(config)#
```

username sshkey

The **username sshkey** command configures an ssh key for the specified username. Command options allow the key to be entered directly into the CLI or referenced from a file.

The specified username must be previously configured through a **username** command.

The **no username sshkey** and **default username sshkey** command deletes the sshkey for the specified username by removing the corresponding **username sshkey** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
username name sshkey KEY
no username name sshkey
default username name sshkey
```

Parameters

- *name* username text that the user enters at the login prompt to access the CLI.

  Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters:

  ```
  @   #   $   %   ^   &   *   (   )   -   _   =
  +   {   }   [   ]   ;   <   >   ,   .   ~   |
  ```

- ***KEY*** SSH key. Options include:

  — *key_text* username is associated with ssh key specified by *key_text* string.
  — **file** *key_file* username is associated with ssh key in the specified file.

Example

- These commands create the username *john*, assign it the password *x245,* then associates it to the SSH key listed in the file named ***john-ssh***.

  ```
  switch(config)#username john secret x245
  switch(config)#username john sshkey file john-ssh
  switch(config)#
  ```

Chapter 5

Administering the Switch

This chapter describes administrative tasks that are typically performed only after initially configuring the switch or after recovery procedures.

This chapter includes these sections:

- Section 5.1: Managing the Switch Name
- Section 5.2: Managing the System Clock
- Section 5.3: Managing Display Attributes
- Section 5.4: Event Monitor
- Section 5.5: Switch Administration Commands

5.1 Managing the Switch Name

These sections describe how to configure the switch's domain and host name.

- Section 5.1.1: Assigning a Name to the Switch describes the assigning of an FQDN to the switch.
- Section 5.1.2: Specifying DNS Addresses describes the adding of name servers to the configuration.

5.1.1 Assigning a Name to the Switch

A fully qualified domain name (FQDN) labels the switch and defines its organization ID in the Domain Name System hierarchy. The switch's FQDN consists of a host name and domain name.

The host name is uniquely associated with one device within an IP-domain. The default host name is ***localhost***. You can configure the prompt to display the host name, as described in Section 5.3.2: Prompt.

- To assign a host name to the switch, use the **hostname** command. To return the switch's host name to the default value of ***localhost***, use the **no hostname** command.
- To specify the domain location of the switch, use the **ip domain-name** command.

  Examples

  - This command assigns the string *main-host* as the switch's host name. The prompt was previously configured to display the host name.

    ```
    Switch(config)#hostname main-host
    main-host(config)#
    ```

  - This command configures *aristanetworks.com* as the switch's domain name.

    ```
    Switch(config)#ip domain-name aristanetworks.com
    Switch(config)#
    ```

- This procedure configures ***sales1.samplecorp.org*** as the switch's FQDN.

```
Switch(config)#hostname sales1
sales1(config)#ip domain-name samplecorp.org
sales1(config)#
```

- This ***running-config*** extract contains the switch's host name and IP-domain name.

```
main-host#show running-config
! device: main-host (DCS-7124S, EOS-4.5.0-010707.2010gaganemgr44)
!
vlan 3-4
!
username john secret 5 $1$a7Hjept9$TIKRX6ytkg8o.ENja.na50
!
hostname sales1
ip name-server 172.17.0.22
ip domain-name samplecorp.org
!
<-------OUTPUT OMITTED FROM EXAMPLE-------->
!
end
main-host#
```

5.1.2 Specifying DNS Addresses

The Domain Name Server (DNS) maps FQDN labels to IP addresses and provides addresses for network devices. Each network requires at least one server to resolve addresses. The configuration file can list a maximum of three server addresses.

To add name servers to the configuration, use the **ip name-server** command. Each command can add one to three servers. The switch disregards any attempt to add a fourth server to the configuration. All server addresses must be in a single VRF. If servers have been previously configured in a different VRF they must be removed before adding a new server to the configuration.

Example

- This code performs these actions:
  - adds three names servers to the configuration in the default VRF
  - attempts to add a fourth server, resulting in an error message
  - displays the configuration file.

```
Switch(config)#ip name-server 10.1.1.24 10.1.1.25 172.17.0.22
Switch(config)#ip name-server 10.15.3.28
% Maximum number of nameservers reached. '10.15.3.28' not added
Switch(config)#show running-config
! device: Switch (DCS-7124S, EOS-4.5.0-236707.2010gaganemgr44 (engineering
build))
!
username david secret 5 $1$a7Hjept9$TIKRX6ytkg8o.ENja.na50
!
hostname Switch
ip name-server 10.1.1.24
ip name-server 10.1.1.25
ip name-server 172.17.0.22
ip domain-name aristanetworks.com
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

5.2 Managing the System Clock

The switch uses the system clock for displaying the time and time-stamping messages. The switch supports time updates through an NTP server or through CLI commands.

The system clock is set to Coordinated Universal Time (UTC), also known as Greenwich Mean Time (GMT). The switch determines local time through time zone commands. Time-stamps and time displays are in local time.

5.2.1 Configuring the Time Zone

The time zone setting converts the system time (UTC) to local time. To specify the time zone, use the **clock timezone** command.

Examples

- These commands configure the switch for the United States Central Time Zone.

```
Switch(config)#clock timezone US/Central
Switch(config)#show clock
Fri Apr 23 18:42:49 2010
timezone is US/Central
Switch(config)#
```

- To view the predefined time zone labels, enter **clock timezone** with a question mark.

```
Switch(config)#clock timezone ?
  Africa/Abidjan                  Africa/Accra
  Africa/Addis_Ababa              Africa/Algiers
<-------OUTPUT OMITTED FROM EXAMPLE-------->
  W-SU                            W-SU timezone
  WET                             WET timezone
  Zulu                            Zulu timezone

Switch(config)#clock timezone
```

- This command displays all time zone labels that start with **America**.

```
Switch(config)#clock timezone AMERICA?
America/Adak                    America/Anchorage
America/Anguilla                America/Antigua
<-------OUTPUT OMITTED FROM EXAMPLE-------->
America/Winnipeg                America/Yakutat
America/Yellowknife

Switch(config)#clock timezone AMERICA
```

5.2.2 Configuring NTP

Network Time Protocol (NTP) servers synchronize time settings of systems running an NTP client. After configuring the switch to synchronize with an NTP server, it may take up to ten minutes for the switch to set its clock. The *running-config* lists NTP servers that the switch can use.

The **ntp server** command adds a server to the list or modifies the parameters of a previously listed address. When the system contains multiple NTP servers, the *prefer* keyword determines the primary NTP server; otherwise, the switch selects servers in their order in *running-config* file. All NTP servers must be in the same VRF. Servers are added in the main VRF if no VRF is specified.

The **ntp bind** command specifies an interface for accessing the IP address of the NTP server as configured by the **ntp server** command. This command is required when the switch configuration

contains more than 1023 IP addresses. *Running-config* can contain multiple **ntp bind** commands. An **ntp bind** command takes effect if the specified interface and the NTP server both belong to the main VRF.

The **ntp source** command configures an interface as the source of NTP packets. The IP address of the interface is used as the source address for all packets sent to all destinations unless a server-specific source is configured using the **source** option of the **ntp server** command. For an **ntp source** command to take effect, the specified interface and the NTP server must both belong to the main VRF.

These commands display the status of the switch NTP server connections:

- **show ntp status**
- **show ntp associations**

Examples

- These commands add three NTP servers, designating the second server as the primary.

```
Switch(config)#ntp server local-NTP
Switch(config)#ntp server 172.16.0.23 Prefer
Switch(config)#ntp server 172.16.0.25
```

- This command displays the status of an NTP connection.

```
Switch(config)#show ntp status
unsynchronised
  time server re-starting
   polling server every 64 s
```

- This command displays data about the NTP servers in the configuration.

```
Switch(config)#show ntp associations
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 moose.aristanet 66.187.233.4     2 u    9   64  377    0.118  9440498    0.017
 172.17.2.6      .INIT.          16 u    - 1024    0    0.000    0.000    0.000
*LOCAL(0)        .LOCL.          10 l   41   64  377    0.000    0.000    0.000
```

5.2.3 Setting the System Clock Manually

The **clock set** command manually configures the system clock time and date, in local time. NTP servers override time that is manually entered.

Example

- This command manually sets the switch time.

```
Switch#clock set 08:15:24 26 April 2010
Mon Apr 26 08:15:25 2010
timezone is US/Central
Switch#
```

5.2.4 Displaying the Time

To display the local time and configured time zone, enter the **show clock** command.

Example

- This command displays the switch time.

```
Switch(config)>show clock
Fri Apr 23 16:32:46 2010
timezone is America/Los_Angeles
Switch(config)>
```

5.3 Managing Display Attributes

Display commands control banner and the command line prompt content.

5.3.1 Banners

The switch can display two banners:

- **Login banner**: The login banner precedes the login prompt. One common use for a login banner is to warn against unauthorized network access attempts.
- **motd banner**: The message of the day (motd) banner is displayed after a user logs into the switch.

This output displays both banners in bold:

```
This is a login banner
Switch login: john
Password:
Last login: Mon Apr 26 09:24:36 2010 from adobe-wrks.aristanetworks.com
This is an motd banner
Switch>
```

These commands create the login and motd banner shown earlier in this section.

```
Switch(config)#banner login
Enter TEXT message. Type 'EOF' on its own line to end.
This is a login banner
EOF
Switch(config)#banner motd
Enter TEXT message. Type 'EOF' on its own line to end.
This is an motd banner
EOF
Switch(config)#
```

To create a banner:

Step 1 Enter Global Configuration mode.

```
Switch#config
Switch(config)#
```

Step 2 Enter banner edit mode by typing the desired command:

- To create a login banner, type **banner login**.
- To create a motd banner, type **banner motd**.

The switch responds with instructions on entering the banner text.

```
Switch(config)#banner login
Enter TEXT message. Type 'EOF' on its own line to end.
```

Step 3 Enter the banner text.

```
This is the first line of banner text.
This is the second line of banner text.
```

Step 4 Press **Enter** to place the cursor on a blank line after completing the banner text.

Step 5 Exit banner edit mode by typing EOF.

```
EOF
Switch(config)#
```

5.3.2 Prompt

The prompt provides an entry point for EOS commands. The **prompt** command configures the contents of the prompt. The **no prompt** command returns the prompt to the default of %H%P.

Characters allowed in the prompt include A-Z, a-z, 0-9, and these punctuation marks:

! @ # $ % ^ & * ( ) - = + f g [ ] ; : < > , . ? / ~ n

The prompt supports these control sequences:

- %s – space character
- %t – tab character
- %% – percent character
- %H – host name
- %D – time and date
- %D{*f_char*} – time and date, format specified by the BSD **strftime (*f_char*)** time conversion function.
- %h – host name up to the first '.'
- %P – extended command mode
- %p – command mode
- %r[1] – redundancy status on modular systems
- %R[2] – extended redundancy status on modular systems – includes status and slot number

Examples

- This command creates a prompt that displays **system 1** and the command mode.

```
host-name.dut103(config)#prompt system%s1%P
system 1(config) #
```

- This command creates a prompt that displays the command mode.

```
host-name.dut103(config)#prompt %p
(config)#
```

- These equivalent commands create the default prompt.

```
% prompt %H%P
host-name.dut103(config)#


% no prompt
host-name.dut103(config)#
```

1. When logged into a fixed system or a supervisor on a modular system, this option has no effect.
2. When logged into a fixed system, this option has no effect.

5.4 Event Monitor

The event monitor writes system event records to local files for access by SQLite database commands.

5.4.1 Description

The event monitor receives notifications for changes to the mac table, route table, and arp table. These changes are logged to a fixed-size circular buffer. The size of this buffer is configurable, but it does not grow dynamically. Buffer contents can be stored to permanent files to increase the event monitor effective capacity. The permanent file size and the number of permanent files is configurable. The buffer is stored at a fixed location on the switch. The location of the permanent files is configurable and can be in any switch file directory, including flash (**/mnt/flash**).

Specific event monitor queries are available through CLI commands. For queries not available through specific commands, manual queries are supported through other CLI commands. When the user issues a query command, the relevant events from the circular buffer and permanent files are written to and accessed from a temporary SQLite database file. The database keeps a separate table for each logging type (mac, arp, route). When the monitor receives notification of a new event, the database file is deleted, then recreated.

5.4.2 Configuring the Event Monitor

Enabling the Event Monitor

The **event-monitor <log enable>** command enables the event monitor and specifies the types of events that are logged. The event monitor is an event logging service that records system events to a local database. The event monitor records these events:

- mac changes to the MAC address table containing (MAC address to port mappings).
- route changes to the IP routing table
- arp changes to the ARP table (IP address to MAC address mappings).

By default, the event monitor is enabled and records each type of event. The **no event-monitor all** disables the event monitor. The **no event-monitor** command, followed by a log type parameter, disables event recording for the specified type.

Example

- This command disables the event monitor for all types of events.

```
Switch(config)#no event-monitor all
```

- This command enables the event monitor for routing table changes.

```
Switch(config)#event-monitor route
```

The **event-monitor clear** command removes the contents of the event monitor buffer. If event monitor backup is enabled, this command removes the contents from all event monitor backup files.

Example

- This command clears the contents of the event monitor buffer.

```
Switch#event-monitor clear
```

Configuring the Buffer

The **event-monitor buffer max-size** command specifies the size of the event monitor buffer. The event monitor buffer is a fixed-size circular data structure that receives event records from the event monitor. When event monitor backup is enabled, the buffer is copied to a backup file before each rollover.

Buffer size ranges from 6 Kb to 50 Kb. The default size is 32 Kb.

- This command configures a buffer size of 48 Kb.

```
Switch(config)#event-monitor buffer max-size 48
```

Configuring Permanent Files

The **event-monitor backup path** command enables the storage of the event monitor buffer to permanent switch files and specifies the path/name of these files. The command references the file location either from the flash drive root directory where the CLI operates (*/mnt/flash*) or from the switch root directory (/).

The event monitor buffer is circular – after the buffer is filled, new data is written to the beginning of the buffer, replacing old data. At the conclusion of each buffer writing cycle, it is copied into a new backup file before the switch starts re-writing the buffer.

Example

- These commands configure the switch to store the event monitor buffer in sw-event.log, then display the new file in the flash directory.

```
Switch(config)#event-monitor backup path sw-event.log
Switch(config)#dir
Directory of flash:/

       -rwx   245761935           Jan 18 04:18  EOS-4.9.0.swi
       -rwx   245729161           Jan 17 06:57  EOS-4.9.0f.swi
       -rwx          25            Jan 5 08:59  boot-config
       -rwx          14           Jun 20  2011  boot-extensions
       -rwx        2749           Nov 22  2011  startup-config
       -rwx      418884           Jan 18 13:55  sw-event.log.0
       -rwx          13            Nov 9  2011  zerotouch-config

931745792 bytes total (190517248 bytes free)
Switch(config)#
```

The **event-monitor backup max-size** command specifies the quantity of event monitor backup files the switch maintains. The switch appends a extension number to the file name when it creates a new file. After every 500 events, the switch deletes the oldest backup file if the file limit is exceeded.

Example

- These commands configures the switch backup the event buffer to a series of files named sw-event.log. The switch can store a maximum of four files.

```
Switch(config)#event-monitor backup path sw-event.log
Switch(config)#event-monitor backup max-size 4
Switch(config)#
```

The first five files that the switch creates to store event monitor buffer contents are:

sw-event.log.0
sw-event.log.1
sw-event.log.2
sw-event.log.3
sw-event.log.4

The switch deletes *sw-event.log.0* the first time it verifies the number of existing backup files after the creation of *sw-event.log.4*.

5.4.3 Querying the Event Monitor

These CLI commands perform SQL-style queries on the event monitor databse:

- The **show event-monitor arp** command displays ARP table events.
- The **show event-monitor mac** command displays MAC address table events.
- The **show event-monitor route** command displays routing table events.

Example

- This command displays all events triggered by MAC address table events.

```
switch#show event-monitor mac
% Writing 0 Arp, 0 Route, 1 Mac events to the database
2012-01-19 13:57:55|1|08:08:08:08:08:08|Ethernet1|configuredStaticMac|added|0
```

For other database queries, the **show event-monitor sqlite** command performs an SQL-style query on the database, using the statement specified in the command.

Example

- This command displays the status of ports in the two port groups on a DCS-7050Q-16 switch.

```
switch#show event-monitor sqlite select * from route;
2012-01-19 13:53:01|16.16.16.0/24||||removed|0
2012-01-19 13:53:01|16.16.16.17/32||||removed|1
2012-01-19 13:53:01|16.16.16.18/32||||removed|2
2012-01-19 13:53:01|16.16.16.240/32||||removed|5
2012-01-19 13:53:01|16.16.16.0/32||||removed|6
2012-01-19 13:53:01|16.16.16.255/32||||removed|7
2012-01-19 13:53:01|192.168.1.0/24||||removed|8
2012-01-19 13:53:01|192.168.1.5/32||||removed|9
2012-01-19 13:53:01|192.168.1.6/32||||removed|10
```

5.4.4 Accessing Event Monitor Database Records

The **event-monitor interact** command replaces the CLI prompt with an SQLite prompt. The event monitor buffer and all backup logs are synchronized into a single SQLite file and loaded for access from the prompt.

- To access help from the SQLite prompt, enter ***.help***
- To exit SQLite and return to the CLI prompt, enter ***.quit*** or ***.exit***

Example

- This command replaces the EOS CLI prompt with an SQLite prompt.

```
switch#event-monitor interact
sqlite>
```

Example

- This command exits SQLite and returns to EOS CLI prompt.

```
sqlite> .quit
switch#
```

The **event-monitor sync** command combines the event monitor buffer and all backup logs and synchronizes them into a single SQLite file, which is stored at */tmp/eventmon.db*

Example

- This command synchronizes the buffer and backup logs into a single SQLite file.

```
switch(config)#event-monitor sync
```

5.5 Switch Administration Commands

This section contains descriptions of the CLI commands that this chapter references.

Switch Name Configuration Commands

- hostname . . . . . Page 174
- ip domain-name . . . . . Page 175
- ip host . . . . . Page 176
- ip name-server . . . . . Page 177
- ipv6 host . . . . . Page 178
- show hosts . . . . . Page 191
- show ip domain-name . . . . . Page 192
- show ip name-server . . . . . Page 193

Clock Configuration Commands

- clock set . . . . . Page 163
- clock timezone . . . . . Page 164
- ntp bind . . . . . Page 179
- ntp server . . . . . Page 180
- ntp source . . . . . Page 182
- show clock . . . . . Page 186
- show ntp associations . . . . . Page 194
- show ntp status . . . . . Page 195

Banner Configuration Commands

- banner motd . . . . . Page 162
- banner login . . . . . Page 161
- show banner . . . . . Page 185

Prompt Configuration Command

- prompt . . . . . Page 183

Event Manager Commands

- no event-monitor . . . . . Page 166
- event-monitor <log enable> . . . . . Page 167
- event-monitor backup max-size . . . . . Page 168
- event-monitor backup path . . . . . Page 169
- event-monitor buffer max-size . . . . . Page 170
- event-monitor clear . . . . . Page 171
- event-monitor interact . . . . . Page 172
- event-monitor sync . . . . . Page 173
- show event-monitor arp . . . . . Page 187
- show event-monitor mac . . . . . Page 188
- show event-monitor route . . . . . Page 189
- show event-monitor sqlite . . . . . Page 190

Email Configuration Command

- email . . . . . Page 165

banner login

The **banner login** command configures a message that the switch displays before login and password prompts. The login banner is available on console, telnet, and ssh connections.

The **no banner login** and **default banner login** commands delete the login banner.

Command Mode

Global Configuration

Command Syntax

```
banner login
no banner login
default banner login
```

Parameters

- *banner_text* To configure the banner, enter a message when prompted. The message may span multiple lines. Banner text supports the following keywords:
  - **$(hostname)** displays the switch's host name.
- **EOF** To end the banner edit session, type on its own line and press **enter**.

Examples

- These commands create a two-line login banner.

```
switch(config)#banner login
Enter TEXT message. Type 'EOF' on its own line to end.
This is a login banner for $(hostname).
Enter your login name at the prompt.
EOF
switch(config)#
```

This output displays the login banner.

```
This is a login banner for switch.
Enter your login name at the prompt.
switch login: john
Password:
Last login: Mon Apr 26 09:05:23 2010 from adobe-wrks.aristanetworks.com
switch>
```

banner motd

The **banner motd** command configures a “message of the day” (motd) that the switch displays after a user logs in. The motd banner is available on console, telnet, and ssh connections.

The **no banner motd** and **default banner motd** commands delete the motd banner.

Command Mode

Global Configuration

Command Syntax

```
banner motd
no banner motd
default banner motd
```

Parameters

- *banner_text* To configure the banner, enter a message when prompted. The message may span multiple lines. Banner text supports this keyword:
  - **$(hostname)** displays the switch’s host name.
- **EOF** To end the banner edit, type on its own line and press **enter**.

Examples

- These commands create an motd banner.

```
switch(config)#banner motd
Enter TEXT message. Type 'EOF' on its own line to end.
This is an motd banner for $(hostname)
EOF
switch(config)#
```

This output displays the motd banner.

```
switch login: john
Password:
Last login: Mon Apr 26 09:17:09 2010 from adobe-wrks.aristanetworks.com
This is an motd banner for Switch
switch>
```

clock set

The **clock set** command sets the system clock time and date. If the switch is configured with an NTP server, NTP time synchronizations override manually entered time settings.

Time entered by this command is local, as configured by the **clock timezone** command.

Command Mode

Privileged EXEC

Command Syntax

clock set *hh.mm.ss date*

Parameters

- *hh.mm.ss* is the time of day, in 24-hour notation.
- *date* is the current date. Date formats include:
  - mm/dd/yy *example:* 05/15/2010
  - Month day year *example:* May 15 2010
  - day month year *example:* 15 May 2010

Examples

- This command manually sets the switch time.

```
switch#clock set 08:15:24 26 April 2010
Mon Apr 26 08:15:25 2010
timezone is US/Central
```

clock timezone

The **clock timezone** command specifies the UTC offset that converts system time to local time. The switch uses local time for time displays and to time-stamp system logs and messages.

The **no clock timezone** and **default clock timezone** commands delete the **timezone** statement from *running-config*, setting local time to UTC.

Command Mode

Global Configuration

Command Syntax

```
clock timezone zone_name
no clock timezone
default clock timezone
```

Parameters

- *zone_name* the time zone. Settings include a list of predefined time zone labels.

Examples

- This command configures the switch for the United States Central Time Zone.

```
switch(config)#clock timezone US/Central
switch(config)#show clock
Fri Apr 23 18:42:49 2010
timezone is US/Central
switch(config)#
```

- To view the predefined time zone labels, enter **clock timezone** with a question mark.

```
switch(config)#clock timezone ?
  Africa/Abidjan                    Africa/Accra
  Africa/Addis_Ababa                Africa/Algiers
  Africa/Asmara                     Africa/Asmera
  Africa/Bamako                     Africa/Bangui
<-------OUTPUT OMITTED FROM EXAMPLE-------->
  W-SU                              W-SU timezone
  WET                               WET timezone
  Zulu                              Zulu timezone

switch(config)#clock timezone
```

- This command displays all time zone labels that start with **America**.

```
switch(config)#clock timezone AMERICA?
America/Adak                    America/Anchorage
America/Anguilla                America/Antigua
America/Araguaina               America/Argentina/Buenos_Aires
<-------OUTPUT OMITTED FROM EXAMPLE-------->
America/Virgin                  America/Whitehorse
America/Winnipeg                America/Yakutat
America/Yellowknife

switch(config)#clock timezone AMERICA
```

email

The **email** command places the switch in email client configuration mode. If you configure a ***from-user*** and an outgoing SMTP server on the switch, you can then use an email address as an output modifier to a **show** command and receive the output as email.

Command Mode

Global Configuration

Command Syntax

email

Example

- This command places the switch in email client configuration mode.

```
switch(config)#email
switch(config)#
```

no event-monitor

The **no event-monitor** and **default event-monitor** commands remove the specified **event-monitor** configuration statements from ***running-config***, returning the switch to the specified default state.

- **no event-monitor <**with no parameters> restores all default setting states:
  - — event monitor is enabled.
  - — buffer backup is disabled.
  - — buffer size is 32 kb.
- **no event-monitor backup** disables the backup.
- **no event-monitor buffer** restores the buffer to the default size.

To disable the event monitor, enter the **no event-monitor all** command (**event-monitor <log enable>**).

Command Mode

Global Configuration

Command Syntax

```
no event-monitor [PARAMETER]
default event-monitor [PARAMETER]
```

Parameters

- *PARAMETER* the event monitor property that is returned to the default state.
  - — <no parameter> all event monitor properties.
  - — **backup** event monitor buffer backup is disabled.
  - — **buffer** the event monitor buffer is restored to its default size.

Examples

- This command removes all event monitor configuration statements from ***running-config***.

```
switch(config)#no event-monitor
switch(config)#
```

event-monitor <log enable>

The **event-monitor <log enable>** command enables the event monitor and specifies the types of events that are logged. The event monitor is an event logging service that records system events to a local database. The event monitor records these events:

- mac changes to the MAC address table containing (MAC address to port mappings).
- route changes to the IP routing table
- arp changes to the ARP table (IP address to MAC address mappings).

The database maintains a separate table for each event type.

By default, the event monitor is enabled and records each type of event.

- The **no event-monitor all** command disables the event monitor.
- The **no event-monitor** command, followed by a log type parameter, disables event recording for the specified type.
- The **event-monitor** and **default event-monitor** commands enable the specified event logging type by removing the corresponding **no event-monitor** command from ***running-config***.

The **no event-monitor** and **default event-monitor** commands, without any log type parameter, restore the default event monitor settings by deleting all event monitor related commands from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
event-monitor LOG_TYPE
no event-monitor LOG_TYPE
default event-monitor LOG_TYPE
```

Parameters

- *LOG_TYPE* specifies the event logging type. Options include:
  - **all** all event logging types.
  - **arp** changes to ARP table.
  - **mac** changes to MAC address table.
  - **route** changes to IP routing table.

Related Commands

- **no event-monitor**

Examples

- This command disables the event monitor for all types of events.

```
switch(config)#no event-monitor all
switch(config)#
```

- This command enables the event monitor for routing table changes.

```
switch(config)#event-monitor route
switch(config)#
```

event-monitor backup max-size

The **event-monitor backup max-size** command specifies the quantity of event monitor backup files the switch maintains. Values range from 1 to 200 files with a default of ten files.

The **event-monitor backup path** command specifies the path/name of these files. The switch appends an extension to the file name that tracks the creation order of backup files. When the quantity of files exceeds the configured limit, the switch deletes the oldest file.

The **no event-monitor backup max-size** and **default event-monitor backup max-size** command restores the default maximum number of backup files the switch can store to ten by removing the corresponding **event-monitor backup max-size** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
event-monitor backup max-size file_quantity
no event-monitor backup max-size
default event-monitor backup max-size
```

Parameters

- *file_quantity* maximum number of backup files. Value ranges from 1 to 200. Default is 10.

Examples

- These commands configures the switch backup the event buffer to a series of files named sw-event.log. The switch can store a maximum of four files.

```
switch(config)#event-monitor backup path sw-event.log
switch(config)#event-monitor backup max-size 4
switch(config)#
```

The first five files that the switch creates to store event monitor buffer contents are:

sw-event.log.0
sw-event.log.1
sw-event.log.2
sw-event.log.3
sw-event.log.4

The switch deletes ***sw-event.log.0*** the first time it verifies the number of existing backup files after the creation of ***sw-event.log.4***.

event-monitor backup path

The **event-monitor backup path** command enables the storage of the event monitor buffer to switch files and specifies the path/name of these files. The command references the file location either from the flash drive root directory (/mnt/flash) where the CLI operates or from the switch root directory (/).

The event monitor buffer is circular – after the buffer is filled, new data is written to the beginning of the buffer, replacing old data. At the conclusion of each buffer writing cycle, it is copied into a new backup file before the switch starts re-writing the buffer. The switch appends a extension number to the file name when it creates a new file. After every 500 events, the switch deletes the oldest backup file if the file limit specified by the **event-monitor backup max-size** command is exceeded.

running-config can contain a maximum of one **event-monitor backup path** statement . Subsequent **event-monitor backup path** commands replace the existing statement in ***running-config***, changing the name of the file where event monitor backup files are stored.

The **no event-monitor backup path** and **default event-monitor backup path** commands disable the storage of the event monitor buffer to switch files by deleting the **event-monitor backup path** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
event-monitor backup path URL_FILE
no event-monitor backup path
default event-monitor backup path
```

Parameters

- ***URL_FILE*** path and file name of the backup file
  - — *path_string* specified path is appended to /mnt/flash/
  - — **file:** *path_string* specified path is appended to /
  - — **flash:** *path_string* specified path is appended to /mnt/flash/

Examples

- These commands configure the switch to store the event monitor buffer in sw-event.log, then displays the new file in the flash directory.

```
switch(config)#event-monitor backup path sw-event.log
switch(config)#dir
Directory of flash:/

       -rwx   245761935           Jan 18 04:18  EOS-4.9.0.swi
       -rwx   245729161           Jan 17 06:57  EOS-4.9.0f.swi
       -rwx          25            Jan 5 08:59  boot-config
       -rwx          14           Jun 20  2011  boot-extensions
       -rwx        2749           Nov 22  2011  startup-config
       -rwx      418884           Jan 18 13:55  sw-event.log.0
       -rwx          13            Nov 9  2011  zerotouch-config

931745792 bytes total (190517248 bytes free)
switch(config)#
```

event-monitor buffer max-size

The **event-monitor buffer max-size** command specifies the size of the event monitor buffer. The event monitor buffer is a fixed-size circular data structure that receives event records from the event monitor. When event monitor backup is enabled (**event-monitor backup path**), the buffer is copied to a backup file before each rollover.

Buffer size ranges from 6 Kb to 50 Kb. The default size is 32 Kb.

The **no event-monitor buffer max-size** and **default event-monitor buffer max-size** command restores the default buffer size of 32 Kb by removing the **event-monitor buffer max-size** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
event-monitor buffer max-size buffer_size
no event-monitor buffer max-size
default event-monitor buffer max-size
```

Parameters

- *buffer_size* buffer capacity (Kb). Values range from 6 to 50. Default value is 32.

Examples

- This command configures a buffer size of 48 Kb.

```
switch(config)#event-monitor buffer max-size 48
switch(config)#
```

event-monitor clear

The **event-monitor clear** command removes the contents of the event monitor buffer. If event monitor backup is enabled, this command removes the contents from all event monitor backup files.

Command Mode

Privileged EXEC

Command Syntax

```
event-monitor clear
```

Examples

- This command clears the contents of the event monitor buffer.

```
switch#event-monitor clear
switch#
```

event-monitor interact

The **event-monitor interact** command replaces the CLI prompt with an SQLite prompt. The event monitor buffer and all backup logs are synchronized into a single SQLite file and loaded for access from the prompt.

- To access help from the SQLite prompt, enter ***.help***
- To exit SQLite and return to the CLI prompt, enter ***.quit*** or ***.exit***

Command Mode

Privileged EXEC

Command Syntax

```
event-monitor interact
```

Examples

- This command replaces the EOS CLI prompt with an SQLite prompt.

```
switch#event-monitor interact
sqlite>
```

- This command exits SQLite and returns to EOS CLI prompt.

```
sqlite> .quit
switch#
```

event-monitor sync

The **event-monitor buffer sync** command combines the event monitor buffer and all backup logs and synchronizes them into a single SQLite file, which is stored at */tmp/eventmon.db*

Command Mode

Privileged EXEC

Command Syntax

```
event-monitor sync
```

Example

- This command synchronizes the buffer and backup logs into a single SQLite file.

```
switch(config)#event-monitor sync
switch(config)#
```

hostname

The **hostname** command assigns a text string as the switch's host name. The default host name is *localhost*.

The prompt displays the host name when appropriately configured through the **prompt** command.

The **no hostname** and **default hostname** commands return the switch's host name to the default value of ***localhost***.

Command Mode

Global Configuration

Command Syntax

hostname *string*
no hostname
default hostname

Parameters

- *string* host name assigned to the switch.

Example

- This command assigns the string *main-host* as the switch's host name.

```
switch(config)#hostname main-host
main-host(config)#
```

The prompt was previously configured to display the host name.

ip domain-name

The **ip domain-name** command configures the switch's domain name. The switch uses this name to complete unqualified host names.

The **no ip domain-name** and **default ip domain-name** commands delete the domain name by removing the **ip domain-name** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip domain-name string
no ip domain-name
default ip domain-name
```

Parameters

- *string* domain name (text string)

Examples

- This command configures *aristanetworks.com* as the switch's domain name.

```
switch(config)#ip domain-name aristanetworks.com
switch(config)#
```

ip host

The **ip host** command associates a hostname to an IP address. This command supports local hostname resolution based on local hostname-IP address maps. Multiple hostnames can be mapped to an IP address. IPv4 and IPv6 addresses can be mapped to the same hostname (**ipv6 host**). The **show hosts** command displays the local hostname-IP address mappings.

The **no ip host** and **default ip host** commands removes hostname-IP address maps by deleting the corresponding **ip host** command from ***running-config***, as specified by command parameters:

- no parameters: command removes all hostname-IP address maps.
- ***hostname*** parameter: command removes all IP address maps for the specified hostname.
- ***hostname*** and ***IP address*** parameters: command removes specified hostname-IP address maps.

Command Mode

Global Configuration

Command Syntax

```
ip host hostname hostadd_1 [hostadd_2] ... [hostadd_X]
no ip host [hostname] [hostadd_1] [hostadd_2] [hostadd_X]
default ip host [hostname] [hostadd_1] [hostadd_2] [hostadd_X]
```

Parameters

- *hostname* hostname (text).
- *hostadd_N* IPv4 address associated with hostname (dotted decimal notation).

Related Commands

- **ipv6 host**
- **show hosts**

Examples

- This command associates the hostname ***test_lab*** with the IP addresses ***10.24.18.5*** and ***10.24.16.3.***

```
switch(config)#ip host test_lab 10.24.18.5 10.24.16.3
```

- This command removes all IP address maps for the hostname ***production_lab***.

```
switch(config)#no ip host production_lab
switch(config)#
```

ip name-server

The **ip name-server** command adds name server addresses to ***running_config***. The switch uses name servers for name and address resolution. The switch can be configured with up to three name servers. Although a command can specify multiple name server addresses, ***running_config*** stores each address in a separate statement. Name server addresses can be IPv4 and IPv6; each command can specify both address types.

Attempts to add a fourth server generate an error message. All name server addresses must be configured in the same VRF. When name servers were previously configured in a VRF, they must all be removed before adding new name server entries.

The **no ip name-server** and **default ip name-server** commands remove specified name servers from ***running_config***. Commands that do not list an address remove all name servers.

Command Mode

Global Configuration

Command Syntax

```
ip name-server [VRF_INSTANCE] SERVER_1 [SERVER_2] [SERVER_3]
no ip name-server [VRF_INSTANCE] [SERVER_1] [SERVER_2] [SERVER_3]
default ip name-server [VRF_INSTANCE] [SERVER_1] [SERVER_2] [SERVER_3]
```

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance containing the addresses.
  - <no parameter> main VRF.
  - **vrf** *vrf_name* a user-defined VRF.
- ***SERVER_X*** IP address of the name server (dotted decimal notation). Options include:
  - *ipv4_addr* (A.B.C.D)
  - *ipv6_addr* (A:B:C:D:E:F:G:H)

  A command can contain both (IPv4 and IPv6) address types.

Guidelines

All configured name server addresses must come from the same VRF. To use a user defined VRF for connection to a name server, first remove any name servers configured in the main VRF.

Examples

- This command adds two name servers to the configuration.

```
switch(config)#ip name-server 172.0.14.21 3:4F21:1902::
switch(config)#
```

- This command attempts to add a name server when the configuration already lists three servers.

```
switch(config)#ip name-server 172.1.10.22
% Maximum number of nameservers reached. '172.1.10.22' not added
switch(config)#
```

ipv6 host

The **ipv6 host** command associates a hostname to an IPv6 address. This command supports local hostname resolution based on local hostname-IP address maps. Multiple hostnames can be mapped to an IPv6 address. IPv4 and IPv6 addresses can be mapped to the same hostname (**ip host**). The **show hosts** command displays the local hostname-IP address mappings.

The **no ipv6 host** and **default ipv6 host** commands removes hostname-IP address maps by deleting the corresponding **ipv6 host** command from ***running-config***, as specified by command parameters:

- no parameters: command removes all hostname-IPv6 address maps.
- ***hostname*** parameter: command removes all IPv6 address maps for the specified hostname.
- ***hostname*** and ***IP address*** parameters: command removes specified hostname-IP address maps.

Command Mode

Global Configuration

Command Syntax

```
ipv6 host hostname hostadd_1 [hostadd_2] ... [hostadd_X]
no ipv6 host [hostname] [hostadd_1] [hostadd_2] [hostadd_X]
default ipv6 host [hostname] [hostadd_1] [hostadd_2] [hostadd_X]
```

Parameters

- *hostname* hostname (text).
- *hostadd_N* IPv6 addresses associated with hostname (dotted decimal notation).

Related Commands

- **ip host**
- **show hosts**

Example

- This command associates the hostname ***support_lab*** with the IPv6 address ***10:14:b2:e9:24:18:93:18.***

```
switch(config)#ipv6 host support_lab 10:14:b2:e9:24:18:93:18
switch(config)#
```

ntp bind

The **ntp bind** command specifies an interface for accessing the IP address of the NTP server as configured by the **ntp server** command. This command is required when the switch configuration contains more than 1023 IP addresses. ***Running-config*** can contain multiple **ntp bind** commands.

The **no ntp bind** and **default ntp bind** commands remove the corresponding **ntp bind** statement from *running-config*. If the command does not specify an interface, all **ntp bind** statements are removed from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
ntp bind INTERFACE_NAME
no ntp bind [INTERFACE_NAME]
default ntp bind [INTERFACE_NAME]
```

Parameters

- *INTERFACE_NAME* interface used for accessing the NTP server address. Options include:
  - **ethernet** *e_range* Ethernet interface list.
  - **loopback** *l_range* loopback interface list.
  - **management** *m_range* management interface list.
  - **port-channel** *c_range* port channel interface list.
  - **vlan** *v_range* VLAN interface list.

  Valid *e_range*, *l_range*, *m_range*, *c_range*, and *v_range* formats include a number, number range, or comma-delimited list of numbers and ranges.

Guidelines

The interface specified in the **ntp bind** command must belong to the main VRF. When NTP servers are configured in a non-default VRF, the **ntp bind** command does not effect the behavior of the switch.

Example

- This command configures the switch to access the NTP server through the Ethernet 7 interface.

```
switch(config)#ntp bind ethernet 7
switch(config)#
```

ntp server

The **ntp server** command adds a Network Time Protocol (NTP) server to ***running-config***. Commands that specify a server that exists in ***running-config*** modify the server settings. The switch synchronizes the system clock with an NTP server when ***running-config*** contains at least one valid NTP server.

The switch supports NTP versions 1 through 4. The default is version 4.

The ***prefer*** option specifies the primary server, giving it higher priority for synchronizing time. If ***running-config*** contains multiple servers with identical priority, the switch uses the first listed server.

The **no ntp server** and **default ntp server** commands remove the specified NTP server from ***running-config***. To remove an NTP server configured in a user defined VRF, include the VRF name in the **no** form of the command. The **no ntp** and **default ntp** commands remove all NTP servers from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ntp server [VRF_INSTANCE] SERVER_NAME [prefer] [NTP_VERSION] [IP_SOURCE]
[burst] [iburst] [MAX_POLL_INT] [MIN_POLL_INT]
no ntp [server [VRF_INSTANCE] SERVER_NAME]
default ntp [server [VRF_INSTANCE] SERVER_NAME]
```

Parameters

- ***VRF_INSTANCE*** the VRF instance to be used for connection to the specified server.
  - <no parameter> connects using the main VRF.
  - **vrf** *vrf_name* connects using the specified user-defined VRF.
- ***SERVER_NAME*** NTP server location. Options include:
  - *IP address* in dotted decimal notation
  - an FQDN host name
- **prefer** indicates the server has priority when the switch selects a synchronizing server.
- ***NTP_VERSION*** specifies the NTP version. Settings include:
  - <no parameter> sets NTP version to 4 (default).
  - **version** *number*, where *number* ranges from 1 to 4.
- ***IP_SOURCE*** specifies the source interface for NTP updates for the specified NTP server. This option overrides global settings created by the **ntp source** command. Options include:
  - <no parameter> sets the source interface to the global default.
  - **source ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **source loopback** *l_num* loopback interface specified by *l_num*.
  - **source management** *m_num* management interface specified by *m_num*.
  - **source port-channel** *p_num* port-channel interface specified by *p_num*.
  - **source vlan** *v_num* VLAN interface specified by *v_num*.
- **burst** indicates that when the NTP server is reached, the switch sends packets to the server in bursts of eight instead of the usual one. Recommended only for local servers. Off by default.
- **iburst** indicates that the switch sends packets to the server in bursts of eight instead of the usual one until the server is reached. Recommended for general use to speed synchronization. Off by default.

- *MAX_POLL_INT* specifies the maximum polling interval for the server. Settings include:
  - <no parameter> sets the maximum polling interval to 10 (1,024 seconds, the default).
  - **maxpoll** *number*, where *number* is the base-2 logarithm of the interval in seconds. Values range from 3 (8 seconds) to 17 (131,072 seconds, approximately 36 hours).
- *MIN_POLL_INT* specifies the minimum polling interval for the server. Settings include:
  - <no parameter> sets the minimum polling interval to 6 (64 seconds, the default).
  - **minpoll** *number*, where *number* is the base-2 logarithm of the interval in seconds. Values range from 3 (8 seconds) to 17 (131,072 seconds, approximately 36 hours).

Guidelines

To configure multiple parameters for a single server, include them all in a single **ntp server** command. Using the command again for the same server overwrites parameters previously configured in ***running-config***.

All NTP servers must use the same VRF. If no VRF is specified, the server is configured in the main VRF. To use a user defined VRF for connection to an NTP server, first use the **no** form of the **ntp server** command to remove any NTP servers configured in the main VRF.

When specifying a source interface, choose an interface in the same VRF as the server. If the source interface is not in the same VRF, the source data will be included in ***running-config*** but will not be added to NTP packets.

An NTP server may be configured using an invalid or inactive VRF, but the status of the NTP server will remain inactive until the VRF is active.

Examples

- This command configures the switch to update its time with the NTP server at address 172.16.0.23 and designates it as a preferred NTP server.

```
switch(config)#ntp server 172.16.0.23 prefer
```

- This command configures the switch to update its time through an NTP server named *local-nettime*.

```
switch(config)#ntp server local-nettime
```

- This command configures the switch to update its time through a version 3 NTP server.

```
switch(config)#ntp server 171.18.1.22 version 3
```

- These commands reconfigure the switch to access the above NTP servers through VRF “magenta”.

```
switch(config)#no ntp server 172.16.0.23
switch(config)#no ntp server local-nettime
switch(config)#no ntp server 171.18.1.22
switch(config)#ntp server vrf magenta 172.16.0.23 prefer
switch(config)#ntp server vrf magenta local-nettime
switch(config)#ntp server vrf magenta 171.18.1.22 version 3
```

ntp source

The **ntp source** command configures an interface as the source of NTP updates. The IP address of the interface is used as the source address for all NTP packets sent to all destinations unless a server-specific source interface has been specified using the **source** option of the **ntp server** command.

The **no ntp source** and **default ntp source** commands remove the **ntp source** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
ntp source INT_PORT
no ntp source
default ntp source
```

Parameters

- *INT_PORT* the interface port that specifies the NTP source. Settings include:
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Guidelines

The **ntp source** command should only be used to specify an interface in the main VRF. To configure an NTP source address using an interface in a user defined VRF, use the **source** option of the **ntp server** command.

Examples

- This command configures VLAN interface 25 as the source of NTP update packets.

```
switch(config)#ntp source vlan 25
switch(config)#
```

- This command removes the NTP source command from the configuration.

```
switch(config)#no ntp source
switch(config)#
```

prompt

The **prompt** command specifies the contents of the CLI prompt. Characters allowed in the prompt include A-Z, a-z, 0-9, and these punctuation marks:

! @ # $ % ^ & * ( ) - = + f g [ ] ; : < > , . ? / ~ n

The prompt supports these control sequences:

- %s – space character
- %t – tab character
- %% – percent character
- %D – time and date
- %D{*f_char*} – time and date, format specified by the BSD **strftime (*f_char*)** time conversion function.
- %H – host name
- %h – host name up to the first '.'
- %P – extended command mode
- %p – command mode
- %r[1] – redundancy status on modular systems
- %R[2] – extended redundancy status on modular systems – includes status and slot number

Table 5-1 displays Command Mode and Extended Command Mode prompts for various modes.

Command Mode	Command Mode Prompt	Extended Command Mode Prompt
Exec	>	>
Privileged Exec	#	#
Global Configuration	(config)#	(config)#
Ethernet Interface Configuration	(config-if)#	(config-if-ET15)#
VLAN Interface Configuration	(config-if)#	(config-if-Vl24)#
Port Channel Interface Configuration	(config-if)#	(config-if-Po4)#
Management Interface Configuration	(config-if)#	(config-if-Ma1)
Access List Configuration	(config-acl)#	(config-acl-listname)#
OSPF Configuration	(config-router)#	(config-router-ospf)#
BGP Configuration	(config-router)#	(config-router-bgp)#

Table 5-1 Command Mode Prompt examples

The **no prompt** and **default prompt** commands return the prompt to the default of %H%R%P.

Command Mode

Global Configuration

Command Syntax

```
prompt p_string
no prompt
default prompt
```

Parameters

- *p_string* prompt text (character string). Elements include letters, numbers, and control sequences.

1. When logged into a fixed system or a supervisor on a modular system, this option has no effect.
2. When logged into a fixed system, this option has no effect.

Examples

- This command creates a prompt that displays **system 1** and the command mode.

```
host-name.dut103(config)#prompt system%s1%P
system 1(config) #
```

- This command creates a prompt that displays the command mode.

```
host-name.dut103(config)#prompt %p
(config)#
```

- These equivalent commands create the default prompt.

```
% prompt %H%P
host-name.dut103(config)#


% no prompt
host-name.dut103(config)#
```

show banner

The **show banner** command displays the specified banner.

Command Mode
Privileged EXEC

Command Syntax
show banner *BANNER_TYPE*

Parameters

- *BANNER_TYPE* banner that the command displays. Options include
  - **login** command displays login banner.
  - **motd** command displays message of the day banner.

Examples

- These commands configure and display the motd banner.

```
switch(config)#banner motd
Enter TEXT message. Type 'EOF' on its own line to end.
This is an motd bannder for $(hostname)
EOF
switch(config)#show banner motd
This is an motd bannder for $(hostname)

switch(config)#
```

show clock

The **show clock** command displays the current system clock time and configured time zone. The switch uses the system clock for system log messages and debugging traces.

Command Mode
EXEC

Command Syntax
show clock

Examples

- This command displays the current system clock time and configured time zone.

```
switch>show clock
Wed Nov  2 10:29:32 2011
timezone is America/Los_Angeles
switch>
```

show event-monitor arp

The **show event-monitor arp** command performs an SQL-style query on the event monitor database and displays ARP table events as specified by command parameters. The event monitor buffer and all backup logs are synchronized into a single SQLite file.

Command Mode

Privileged EXEC

Command Syntax

```
show event-monitor arp [GROUP] [MESSAGES] [INTERFACE] [IP] [MAC] [TIME]
```

Optional parameters can be placed in any order.

Parameters

- ***GROUP*** used with aggregate functions to group results. Analogous to SQL ***group by*** command.
  - <no parameter> results are not grouped.
  - **group-by** *ip* results are grouped by IP address.
  - **group-by** *mac* results are grouped by MAC address.
- ***MESSAGES*** number of message returned from query. Analogous to SQL ***limit*** command.
  - <no parameter> result-set size is not limited.
  - **limit** *msg_quantity* number of results that are displayed. Values range from 1 to 15,000.
- ***INTERFACE*** resticts result-set to events that include specified interface (SQL Like command).
  - <no parameter> result-set not restricted by interface.
  - **match-interface ethernet** *e_range* Ethernet interface list.
  - **match-interface loopback** *l_range* loopback interface list.
  - **match-interface management** *m_range* management interface list.
  - **match-interface port-channel** *c_range* port channel interface list.
  - **match-interface vlan** *v_range* VLAN interface list.
- ***IP*** resticts result-set to events that include specified IP address (SQL Like command).
  - <no parameter> command
  - **match-ip** *ip_address_rex* IP address, as represented by regular expression.
- ***MAC*** resticts result-set to events that include specified MAC address (SQL Like command).
  - <no parameter> command
  - **match-mac** *mac_address_rex* MAC address, as represented by regular expression
- ***TIME*** restricts result-set to events generated during specifed period.
  - <no parameter> result-set not restricted by time of event.
  - **match-time last-minute** includes events generated during last minute.
  - **match-time last-day** includes events generated during last day.
  - **match-time last-hour** includes events generated during last hour.
  - **match-time last-week** includes events generated during last week.

show event-monitor mac

The **show event-monitor mac** command performs an SQL-style query on the event monitor database and displays MAC address table events as specified by command parameters. The event monitor buffer and all backup logs are synchronized into a single SQLite file.

Command Mode

Privileged EXEC

Command Syntax

```
show event-monitor mac [GROUP] [MESSAGES] [INTERFACE] [MAC] [TIME]
```

Optional parameters can be placed in any order.

Parameters

- ***GROUP*** used with aggregate functions to group results. Analogous to SQL ***group by*** command.
  - <no parameter> results are not grouped.
  - **group-by** *interface* results are grouped by interface.
  - **group-by** *mac* results are grouped by MAC address.
- ***MESSAGES*** number of message returned from query. Analogous to SQL ***limit*** command.
  - <no parameter> result-set size is not limited.
  - **limit** *msg_quantity* number of results that are displayed. Values range from 1 to 15,000.
- ***INTERFACE*** resticts result-set to events that include specified interface (SQL Like command).
  - <no parameter> result-set not restricted by interface.
  - **match-interface ethernet** *e_range* Ethernet interface list.
  - **match-interface loopback** *l_range* loopback interface list.
  - **match-interface management** *m_range* management interface list.
  - **match-interface port-channel** *c_range* port channel interface list.
  - **match-interface vlan** *v_range* VLAN interface list.
- ***MAC*** resticts result-set to events that include specified MAC address (SQL Like command).
  - <no parameter> command
  - **match-mac** *mac_address_rex*
- ***TIME*** restricts result-set to events with specifed period (
  - <no parameter> result-set not restricted by time of event.
  - **match-time last-minute** includes events generated during last minute.
  - **match-time last-day** includes events generated during last day.
  - **match-time last-hour** includes events generated during last hour.
  - **match-time last-week** includes events generated during last week.

Examples

- This command displays all events triggered by MAC address table events.

```
switch#show event-monitor mac
% Writing 0 Arp, 0 Route, 1 Mac events to the database
2012-01-19 13:57:55|1|08:08:08:08:08:08|Ethernet1|configuredStaticMac|added|0
```

- This command displays events triggered by MAC address table changes.

```
switch#show event-monitor mac match-mac 08:08:08:%
2012-01-19 13:57:55|1|08:08:08:08:08:08|Ethernet1|configuredStaticMac|added|0
```

show event-monitor route

The **show event-monitor route** command performs an SQL-style query on the event monitor database and displays routing table events as specified by command parameters. The event monitor buffer and all backup logs are synchronized into a single SQLite file.

Command Mode

Privileged EXEC

Command Syntax

```
show event-monitor route [GROUP] [MESSAGES] [IP] [TIME]
```

Optional parameters can be placed in any order.

Parameters

- ***GROUP*** used with aggregate functions to group results. Analogous to SQL ***group by*** command.
  - — <no parameter> results are not grouped.
  - — **group-by** *ip* results are grouped by IP address.
- ***MESSAGES*** number of message returned from query. Analogous to SQL ***limit*** command.
  - — <no parameter> result-set size is not limited.
  - — **limit** *msg_quantity* number of results that are displayed. Values range from 1 to 15,000.
- ***INTERFACE*** resticts result-set to events that include specified interface (SQL Like command).
  - — <no parameter> result-set not restricted by interface.
  - — **match-interface ethernet** *e_range* Ethernet interface list.
  - — **match-interface loopback** *l_range* loopback interface list.
  - — **match-interface management** *m_range* management interface list.
  - — **match-interface port-channel** *c_range* port channel interface list.
  - — **match-interface vlan** *v_range* VLAN interface list.
- ***IP*** resticts result-set to events that include specified IP address (SQL Like command).
  - — <no parameter> command
  - — **match-ip** *ip_address_rex* IP address, as represented by regular expression.
- ***TIME*** restricts result-set to events with specifed period (
  - — <no parameter> result-set not restricted by time of event.
  - — **match-time last-minute** includes events generated during last minute.
  - — **match-time last-day** includes events generated during last day.
  - — **match-time last-hour** includes events generated during last hour.
  - — **match-time last-week** includes events generated during last week.

show event-monitor sqlite

The **show event-monitor sqlite** command performs an SQL-style query on the event monitor database, using the statement specified in the command.

Command Mode
Privileged EXEC

Command Syntax
show event-monitor sqlite *statement*

Parameters

- *statement* SQLite statement.

Examples

- This command displays the status of ports in the two port groups on a DCS-7050Q-16 switch.

```
switch#show event-monitor sqlite select * from route;
2012-01-19 13:53:01|16.16.16.0/24||||removed|0
2012-01-19 13:53:01|16.16.16.17/32||||removed|1
2012-01-19 13:53:01|16.16.16.18/32||||removed|2
2012-01-19 13:53:01|16.16.16.240/32||||removed|5
2012-01-19 13:53:01|16.16.16.0/32||||removed|6
2012-01-19 13:53:01|16.16.16.255/32||||removed|7
2012-01-19 13:53:01|192.168.1.0/24||||removed|8
2012-01-19 13:53:01|192.168.1.5/32||||removed|9
2012-01-19 13:53:01|192.168.1.6/32||||removed|10
```

show hosts

The **show hosts** command displays the default domain name, name lookup service style, a list of name server hosts, and the static hostname-IP address maps.

Command Mode
EXEC

Command Syntax
show hosts

Examples

- This command displays the switch's ip domain name:

```
switch>show hosts

Default domain is: aristanetworks.com
Name/address lookup uses domain service
Name servers are: 172.22.22.40, 172.22.22.10

Static Mappings:

Hostname                         IP     Addresses

TEST_LAB                        IPV4    10.24.18..
PRODUCTION_LAB                  IPV4    24.19.8.31
SUPPORT_LAB                     IPV6    22:49:67:55:18:98:77:64
switch>
```

show ip domain-name

The **show ip domain-name** command displays the switch's IP domain name that is configured with the ip domain name command.

Command Mode

EXEC

Command Syntax

```
show ip domain-name
```

Examples

- This command displays the switch's IP domain name:

```
switch>show ip domain-name
aristanetworks.com
switch>
```

show ip name-server

The **ip name-server** command displays the ip addresses of name-servers in ***running-config***. The name servers are configured by the **ip name-server** command.

Command Mode

EXEC

Command Syntax

show ip name-server

Examples

- This command displays the IP address of name servers that the switch is configured to access.

```
switch>show ip name-server
172.22.22.10
172.22.22.40
switch>
```

show ntp associations

The **show ntp associations** command displays the status of connections to NTP servers.

Command Mode

EXEC

Command Syntax

```
show ntp associations
```

Display Values

- st (stratum): distance from the reference clock
- t (transmission type): u – unicast; b – broadcast; l: local
- when: interval since reception of last packet (seconds unless unit is provided)
- poll: interval between NTP poll packets. Maximum (1024) reached as server and client syncs
- reach: octal number that displays status of last eight NTP messages (377 - all messages received).
- delay: round trip delay of packets to selected reference clock.
- offset: difference between local clock and reference clock.
- jitter: maximum error of local clock relative to reference clock.

Examples

- This command displays the status of the switch's NTP associations.

```
switch>show ntp associations
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 1.1.1.1         .INIT.          16 u    - 1024    0    0.000    0.000   0.000
 moose.aristanet 66.187.233.4     2 u    9   64  377    0.118  9440498   0.017
 172.17.2.6      .INIT.          16 u    - 1024    0    0.000    0.000   0.000
*LOCAL(0)        .LOCL.          10 l   41   64  377    0.000    0.000   0.000
```

show ntp status

The **show ntp status** command displays the NTP parameter settings.

Command Mode
EXEC

Command Syntax
show ntp status

Examples

- This command the switch's NTP parameter settings.

```
switch>show ntp status
synchronised to NTP server (172.22.22.50) at stratum 4
   time correct to within 77 ms
   polling server every 1024 s
switch>
```

Chapter 6

Booting the Switch

This chapter describes the switch boot process, describes configuration options, and lists the components it requires, including the boot loader, the boot loader shell, and other configuration files.

This chapter includes the following sections:

- Section 6.1: Boot Loader – Aboot
- Section 6.2: Configuration Files
- Section 6.3: System Reset
- Section 6.4: Aboot Shell
- Section 6.5: Aboot Configuration Commands
- Section 6.6: Switch Booting Commands

6.1 Boot Loader – Aboot

Aboot is the boot loader for Arista switches. In addition to booting the switch EOS, Aboot provides a shell for changing boot parameters, restoring default switch settings, diagnosing hardware problems, and managing switch files. Section 6.4: Aboot Shell describes the Aboot shell.

The boot process loads an EOS image file, initiates switch processes, performs self tests, restores interface settings, and configures other network parameters. The replacement image file can be in the switch's flash or on a device in the flash drive port. Configuration files stored in flash memory specify boot parameters.

Aboot supports most available USB flash drive models. The flash drive must be formatted with the FAT or VFAT file system. Windows NT File System (NTFS) is not supported.

Aboot initiates a system reboot upon a **reload** command or by restoring power to the switch. Before loading the EOS image file, Aboot provides an option to enter the Aboot shell. The user can either enter the shell to modify boot parameters or allow the switch to boot.

The boot process can be monitored through a terminal connected to the console port. The console port is configured to interact with the terminal by configuration file settings.

6.2 Configuration Files

Three files define boot and running configuration parameters.

- **boot-config:** Contains the location and name of the image to be loaded.
- **running-config**: Contains the current switch configuration.
- **startup-config**: Contains the switch configuration that is loaded when the switch boots.

The ***running-config*** and ***startup-config*** are different when configuration changes have not been saved since the last boot.

6.2.1 boot-config

The ***boot-config*** file is an ASCII file that Aboot uses to configure console communication settings, locate the EOS flash image, and specify initial network configuration settings.

Aboot attempts to boot the EOS flash software image (SWI) referenced by ***boot-config*** if the user does not interrupt the boot process. See Section 6.4: Aboot Shell describes how Aboot uses ***boot-config***.

You can view and edit the ***boot-config*** file contents. Viewing and editing options include:

- View ***boot-config*** file contents with the **more boot-config** command:

```
main-host(config)#more boot-config
SWI=flash:/EOS.swi
CONSOLESPEED=2400
Aboot password (encrypted): $1$A8dZ3GLZ$knKrBpTyg5dhmtGdCdwNM.
main-host(config)#
```

- View ***boot-config*** settings with the **show boot-config** command:

```
main-host(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: 2400
Aboot password (encrypted): $1$A8dZ3GLZ$knKrBpTyg5dhmtGdCdwNM.
main-host(config)#
```

- Modify file settings from the command line with EOS **boot** commands.

  See Section 6.2.1.3: Programming boot-config from the CLI for a list of **boot** commands

- Edit the file directly by using vi from the Bash shell.

  See Section 6.2.1.2: boot-config Command Line Content for a list of ***boot-config*** parameters.

6.2.1.1 boot-config File Structure

Each line in the ***boot-config*** file specifies a configuration setting and has this format:

NAME=VALUE

- NAME is the parameter label.
- VALUE indicates the parameter's bootup setting.

The NAME and VALUE fields cannot contain spaces.

Aboot ignores blank lines and lines that begin with a # character.

6.2.1.2 boot-config Command Line Content

Aboot configuration commands that ***boot-config*** files can contain include:

- **SWI** specifies the location and file name of the EOS image file that Aboot loads when booting, using the same format as the boot command to designate a local or network path.

  Examples

  - `SWI=flash:EOS.swi` (flash drive location)
  - `SWI=usb1:/EOS1.swi` (usb drive location)
  - `SWI=file:/tmp/EOSexp.swi` (switch directory location)
  - `SWI=/mnt/flash/EOS.swi`
  - `SWI=http://foo.com/images/EOS.swi`
  - `SWI=ftp://foo.com/images/EOS.swi`
  - `SWI=tftp://foo.com/EOS.swi`
  - `SWI=nfs://foo.com/images/EOS.swi`

- **CONSOLESPEED** specifies the console baud rate. To communicate with the switch, the connected terminal must match the specified rate. Baud rates are 1200, 2400, 4800, 9600, 19200, or 38400. The default baud rate is 9600.

  Examples

  - CONSOLESPEED=2400
  - CONSOLESPEED=19200

- **PASSWORD (ABOOT)** specifies the Aboot password, as described in Section 6.4.2: Accessing the Aboot Shell. If ***boot-config*** does not contain a PASSWORD line, the Aboot shell does not require a password.

  Examples

  - PASSWORD=1CdWp5wfe$pzNtE3ujBoFEL8vjcq7jo/

- **NET commands** indicate the network interface that ***boot-config*** network settings configure. If ***boot-config*** does not contain a NETDEV setting, the booting process does not attempt to configure a network interface. Other NET commands specify settings that Aboot uses to configure the interface.

  Examples

  - NETDEV command that specifies Ethernet management 1 port.

    ```
    NETDEV=mgmt1
    ```

  - NETAUTO command that configures the interface through a DHCP server, ignoring other NET settings.

    ```
    NETAUTO=dhcp
    ```

  - NET commands that configure the interface manually:

    ```
    NETIP=10.12.15.10
    NETMASK=255.255.255.0
    NETGW=10.12.15.24
    NETDOMAIN=mycompany.com
    NETDNS=10.12.15.13
    ```

6.2.1.3 Programming boot-config from the CLI

The switch CLI provides **boot** commands for editing ***boot-config*** contents. **boot** commands are not accessible from a console port CLI. Parameters not configurable from a boot command can be modified by directly editing the ***boot-config*** file.

Commands that configure boot parameters include boot system, boot secret, and boot console.

boot system

The **boot system** command provides the EOS image file location to Aboot.

Examples

- This command specifies EOS1.swi, on USB flash memory, as the software image load file.

```
main-host(config)#boot system usb1:EOS1.swi
```

  The CLI command places this command in the boot-command file.

```
SWI=usb1:/EOS1.swi
```

- This command designates EOS.swi, on the switch flash, as the EOS software image load file.

```
main-host(config)#boot system flash:EOS.swi
```

  The CLI command places this command in the boot-command file.

```
SWI=flash:/EOS.swi
```

boot secret

The **boot secret** command sets the Aboot password.

Examples

- These equivalent commands set the Aboot password to xr19v:

```
main-host(config)#boot secret xr19v

main-host(config)#boot secret 0 xr19v
```

  This CLI code displays the result:

```
main-host(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): $1$k9YHFW8D$cgM8DSN.e/yY0p3k3RUvk.
```

  The CLI command places this PASSWORD line in the boot-command file.

```
PASSWORD=$1$k9YHFW8D$cgM8DSN.e/yY0p3k3RUvk.
```

  The user must enter xr19v at the login prompt to access the Aboot shell.

- This command sets the Aboot password to ***xr123***. The encrypted string was previously generated with ***xr123*** as the clear text seed.

```
main-host(config)#boot secret 5 $1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
```

  This CLI code displays the result:

```
main-host(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): $1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
```

The CLI command places this PASSWORD line in the boot-command file.

```
PASSWORD=$1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
```

The user must enter ***xr123*** at the login prompt to access the Aboot shell.

- This command removes the Aboot password; subsequent Aboot access is not authenticated.

```
main-host(config)#no boot secret
```

This CLI code displays the result:

```
main-host(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): (not set)
```

boot console

The **boot console** command sets console settings for attaching devices.

Example

- This command sets the console speed to 4800 baud:

```
main-host(config)#boot console speed 4800
```

This CLI code displays the result of the command:

```
main-host(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: 4800
Aboot password (encrypted): (not set)
```

The previous command places this command in the boot-command file.

```
CONSOLESPEED=4800
```

6.2.2 Running-Config

running-config is a virtual file that contains the system's operating configuration, formatted as a command sequence. Commands entered from the CLI modify ***running-config***. Copying a file to ***running-config*** updates the operating configuration by executing the commands in the copied file.

running-config commands include:

- **show running-config** displays ***running-config***.
- **copy running-config startup-config** copies ***running-config*** contents to the ***startup-config***.
- **write memory** copies ***running-config*** contents to the ***startup-config*** file.

6.2.3 Startup-Config

The ***startup-config*** file is stored in flash memory and contains the configuration that the switch loads when booting. During a switch boot, ***running-config*** is replaced by ***startup-config***. Changes to ***running-config*** that are not copied to ***startup-config*** are lost when the system reboots.

startup-config commands include:

- **show startup-config** displays ***startup-config***.
- **copy** <filename> **startup-config** copies contents of the specified file to ***startup-config***.
- **erase startup-config** deletes the ***startup-config*** file.

6.3 System Reset

When a reset condition exists, Aboot can either reset the switch without user intervention or facilitate a manual reset through the Aboot shell. A reset operation clears the switch, including memory states and other hardware logic

- Fixed 1-RU systems: The power supply remains powered up through the reset. Power is removed from all other switch components for two to five seconds.
- Modular systems: The power supply on the active supervisor remains powered up through the reset. Power is removed from all other supervisor components for at least one second. After the supervisor becomes functional, it manages the power-cycling of all line cards.

The **reload** command initiates an immediate reset, terminating all CLI instances not running through the console port. The console port CLI displays messages that the switch generates during a reset.

The **reload <scheduled>** command schedules a reset operation to initiate at a specific time or after a specified period.

6.3.1 Typical Reset Sequence

The **reload** command power cycles the switch, then resets it under Aboot control. The hard reset clears the switch, including memory states and other hardware logic.

By default, the **reload** command triggers a request to store unsaved ***running-config*** commands and an option to open the Aboot shell before starting the reboot. The switch then begins the reboot process controlled by Aboot.

This procedure is an example of a typical restart.

Step 1 Begin the reboot process by typing the **reload** command:

```
main-host#:reload
```

The switch sends a message to confirm the reload request:

```
Proceed with reload? [confirm]
```

Step 2 Press **enter** or type **y** to confirm the requested reload. Pressing any other key terminates the reload operation.

The switch sends a series of messages, including a notification that a message was broadcast to all open CLI instances, informing them that the system is being rebooted. The reload pauses when the CLI displays the Aboot shell notification line.

```
Broadcast message from root@mainStopping sshd: [  OK  ]
SysRq : Remount R/O
Restarting system

Aboot 1.9.0-52504.EOS2.0

Press Control-C now to enter Aboot shell
```

Step 3 To continue the reload process, do nothing. Typing **Ctrl-C** opens the Aboot shell; see Section 6.4.5: Commands for Aboot editing instructions.

The switch continues the reset process, displaying messages to indicate the completion of individual tasks. The reboot is complete when the CLI displays a login prompt.

```
Booting flash:/EOS.swi
Unpacking new kernel
Starting new kernel
Switching to rooWelcome to Arista Networks EOS 4.4.0
Mounting filesystems:  [  OK  ]
Entering non-interactive startup
Starting EOS initialization stage 1: [  OK  ]
ip6tables: Applying firewall rules: [  OK  ]
iptables: Applying firewall rules: [  OK  ]
iptables: Loading additional modules: nf_conntrack_tftp [  OK  ]
Starting system logger: [  OK  ]
Starting system message bus: [  OK  ]
Starting NorCal initialization: [  OK  ]
Starting EOS initialization stage 2: [  OK  ]
Starting ProcMgr: [  OK  ]
Completing EOS initialization: [  OK  ]
Starting Power On Self Test (POST): [  OK  ]
Generating SSH2 RSA host key: [  OK  ]
Starting isshd: [  OK  ]
Starting sshd: [  OK  ]
Starting xinetd: [  OK  ]
[  OK  ] crond: [  OK  ]

main-host login:
```

Step 4 Log into the switch to resume configuration tasks.

6.3.2 Switch Recovery

Aboot can automatically erase the internal flash and copy the contents of a USB key that has been inserted before powering up or rebooting the switch. This recovery method does not require access to the switch console or Aboot password entry, even if the ***boot-config*** file lists one.

Aboot invokes the recovery mechanism only if each of these two conditions is met:

- The USB key must contain a file called **fullrecover**

  The file's contents are ignored; an empty text file is sufficient.

- If the USB key contains a file named ***boot-config***, its timestamp must differ from the timestamp of the ***boot-config*** file on the internal flash.

  This prevents Aboot from invoking the recovery mechanism again on every boot if you leave the flash key inserted.

To use this recovery mechanism, set up a USB key with the files to be installed on the internal flash – for example, a current EOS SWI and a customized or empty ***boot-config*** – plus an empty file named fullrecover.

Check that the timestamp of ***boot-config*** is current to ensure that the above conditions are met.

6.3.3 Display Reload Cause

The **show reload cause** command displays the cause of the most recent system reset and lists recommended actions, if any exist, to avoid future spontaneous resets or resolve other issues that may have cause the reset.

Example

- To display the reset cause, type **show reload cause** at the prompt.

```
main-host: show reload cause
Reload Cause 1:
-------------------
Reload requested by the user.

Recommended Action:
-------------------
No action necessary.

Debugging Information:
----------------------
None available.
localhost#
```

6.3.4 Configuring Zero Touch Provisioning

Zero Touch Provisioning (ZTP) is a switch configuration method that uses files referenced by a DCHP server to initially provision the switch without user intervention. A switch enters ZTP mode when it is reloaded if flash memory does not contain a ***startup-config***.

Cancelling ZTP boots the switch without using a ***startup-config*** file. When ZTP mode is cancelled, a ***startup-config*** file is not stored to flash memory. Until a ***startup-config*** file is stored to flash, the switch returns to ZTP mode on subsequent reboots. This section describes steps required to implement, monitor, and cancel ZTP.

6.3.4.1 Configuring the Network for ZTP

A switch performs the following after booting in ZTP mode:

- Configures each physical interface to ***no switchport*** mode.
- Sends a DHCP query packet on all Ethernet and management interfaces.

After the switch receives a DHCP offer, it responds with a DHCP request for Option 66 (TFTP server name), Option 67 (bootfile name), and dynamic network configuration settings. When the switch receives a valid DHCP response, it configures the network settings, then fetches the file from the location listed in Option 67. If Option 67 returns a network URL (http:// or ftp://), the switch obtains the file from the network. If Option 67 returns a file name, the switch retrieves the file from the TFTP server listed in Option 66.

The Option 67 file can be a ***startup-config*** file or a boot script. The switch distinguishes between a ***startup-config*** file and a boot script by examining the first line in the file:

- The first line of a boot file must consist of the #! characters followed by the interpreter path. The switch executes the code in the script, then reboots. The boot script may fetch an SWI image or perform required customization tasks.

  The following boot file fetches an SWI image and stores a startup configuration file to flash.

```
#!/usr/bin/Cli -p2
copy http://company.com/startup-config flash:startup-config
copy http://company.com/EOS-2.swi flash:EOS-2.swi
config
boot system flash:EOS-2.swi
```

- The switch identifies any other file as a ***startup-config*** file. The switch copies the ***startup-config*** file into flash as ***mnt/flash/startup-config***, then reboots.

The switch uses its system MAC address as the DHCP client identifier and ***Arista*** as the Vendor Class Identifier (Option 60). When the switch receives an http URL through Option 67, it sends the following http headers in the GET request:

```
X-Arista-SystemMAC:
X-Arista-HardwareVersion:
X-Arista-SKU:
X-Arista-Serial:
X-Arista-Architecture:
```

6.3.4.2 Monitoring ZTP Progress

A switch displays the following message after rebooting when it does not contain a ***startup-config*** file:

```
No startup-config was found.

The device is in Zero Touch Provisioning mode and is attempting to
download the startup-config from a remote system. The device will not
be fully functional until either a valid startup-config is downloaded
from a remote system or Zero Touch Provisioning is cancelled. To cancel
Zero Touch Provisioning, login as admin and type 'zerotouch cancel'
at the CLI.

localhost login:
```

The switch displays a CONFIG_DOWNLOAD_SUCCESS message after it successfully downloads a ***startup-config*** file, then continues the reload process as described in Section 6.3.1.

```
================================================================================

Successful download
--------------------


Apr 15 21:36:46 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP request on
[ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18, Ethernet21,
Ethernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9, Management1,
Management2 ]
Apr 15 21:36:56 localhost ZeroTouch: %ZTP-5-DHCP_SUCCESS: DHCP response received
on Ethernet24  [ Mtu: 1500; Ip Address: 10.10.0.4/16; Nameserver: 10.10.0.1;
Domain: aristanetworks.com; Gateway: 10.10.0.1; Boot File:
http://10.10.0.2:8080/tmp/172.17.11.196-startup-config.1 ]
Apr 15 21:37:01 localhost ZeroTouch: %ZTP-5-CONFIG_DOWNLOAD: Attempting to
download the startup-config from
http://10.10.0.2:8080/tmp/172.17.11.196-startup-config.1
Apr 15 21:37:02 localhost ZeroTouch: %ZTP-5-CONFIG_DOWNLOAD_SUCCESS: Successfully
downloaded startup-config from
http://10.10.0.2:8080/tmp/172.17.11.196-startup-config.1
Apr 15 21:37:02 localhost ZeroTouch: %ZTP-5-RELOAD: Rebooting the system
Broadcast messagStopping sshd: [  OK  ]
watchdog is not running
SysRq : Remount R/O
Restarting system
ø

Aboot 1.9.0-52504.EOS2.0


Press Control-C now to enter Aboot shell
```

6.3.4.3 ZTP Failure Notification

The switch displays a DHCP_QUERY_FAIL message when it does not receive a valid DHCP response within 30 seconds of sending the query. The switch then sends a new DHCP query and waits for a response. The switch continues sending queries until it receives a valid response or until ZTP mode is cancelled.

```
localhost login:admin
admin
localhost>Apr 15 21:28:21 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP
request on  [ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18,
Ethernet21, E-thernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9,
Management1, Management2 ]
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-DHCP_QUERY_FAIL: Failed to get a valid
DHCP response
Apr 15 21:28:51 localhost ZeroTouch: %ZTP-5-RETRY: Retrying Zero Touch
Provisioning from the begining (attempt 1)
Apr 15 21:29:22 localhost ZeroTouch: %ZTP-5-DHCP_QUERY: Sending DHCP request on
[ Ethernet10, Ethernet13, Ethernet14, Ethernet17, Ethernet18, Ethernet21,
Ethernet22, Ethernet23, Ethernet24, Ethernet7, Ethernet8, Ethernet9, Management1,
Management2 ]
```

6.3.4.4 Cancelling ZTP Mode

To boot the switch without a ***startup-config*** file, log into the console, then cancel ZTP mode. After the switch boots, it uses all factory default settings. A ***startup-config*** file must be saved to flash memory to prevent the switch from entering ZTP mode on subsequent boots.

See Section 2.1.2.2 for ZTP mode cancellation instructions.

6.3.5 Configuring the Networks

If the ***boot-config*** file contains a NETDEV statement, Aboot attempts to configure the network interface, as specified by Network configuration commands. See Section 6.2.1.2: boot-config Command Line Content for a list of commands that define the network configuration.

6.4 Aboot Shell

The Aboot shell is an interactive command-line interface used to manually boot a switch, restore the internal flash to its factory-default state, run hardware diagnostics, and manage files. The Aboot shell is similar to the Linux Bourne Again Shell (Bash).

The Aboot shell provides commands for restoring the state of the internal flash to factory defaults or a customized default state. You can use these recovery methods to:

- restore the factory-default flash contents before transferring the switch to another owner.
- restore Aboot shell access if the Aboot password is lost or forgotten.
- restore console access if baud rate or other settings are incompatible with the terminal.
- replace the internal flash contents with configuration or image files stored on a USB flash drive.

6.4.1 Operation

When the switch is powered on or rebooted, Aboot reads its configuration from ***boot-config*** on the internal flash and attempts to boot a software image (SWI) automatically if one is configured.

You can monitor the automatic boot process or enter the Aboot shell only from the console port. You can connect a PC or terminal directly to the port and run a terminal emulator to interact with the serial port or access it through a serial concentrator device.

Console settings are stored in ***boot-config***; the factory-default settings for Arista switches are 9600 baud, no parity, 8 character bits, and 1 stop bit. If you do not know the current settings, perform a full flash recovery to restore the factory-default settings. When the console port is connected and the terminal settings are configured properly, the terminal displays a message similar to the following a few seconds after powering up the switch:

```
Aboot 1.0.0

Press Control-C now to enter the Aboot shell
```

To abort the automatic boot process and enter the Aboot shell, press **Ctrl-C** (ASCII 3 in the terminal emulator) after the ***Press Control-C now to enter Aboot shell*** message appears. Pressing **Ctrl-C** can interrupt the boot process up through the starting of the new kernal.

If the ***boot-config*** file does not contain a password command, the Aboot shell starts immediately. Otherwise, you must enter the correct password at the password prompt to start the shell. If you enter the wrong password three times, Aboot displays this message:

```
Type "fullrecover" and press Enter to revert /mnt/flash to factory default
state, or just press Enter to reboot:
```

- Pressing **Enter** continues a normal soft reset without entering the Aboot shell.
- Typing **fullrecover** and pressing **Enter** performs a full flash recovery to restore the factory-default settings, removing all previous contents of the flash drive.

The Aboot shell starts by printing:

```
Welcome to Aboot.
```

Aboot then displays the Aboot# prompt.

Aboot reads its configuration from ***boot-config*** on the internal flash.

6.4.2 Accessing the Aboot Shell

To access the Aboot Shell,

Step 1 Reload the switch and press **enter** or type **y** when prompted, as described by step 1 and step 2 in Section 6.3.1: Typical Reset Sequence.

The command line displays this Aboot entry prompt.

```
Press Control-C now to enter Aboot shell
```

Step 2 Type **Ctrl-C**.

If the ***boot-config*** file does not contain a PASSWORD command, the CLI displays an Aboot welcome banner and prompt.

```
Press Control-C now to enter Aboot shell
^CWelcome to Aboot.
Aboot#
```

If the ***boot-config*** file contains a PASSWORD command, the CLI displays a password prompt. In this case, proceed to step 3. Otherwise, the CLI displays the Aboot prompt.

Step 3 If prompted, enter the Aboot password.

```
Press Control-C now to enter Aboot shell
^CAboot password:
Welcome to Aboot.
Aboot#
```

Aboot allows three attempts to enter the correct password. After the third attempt, the CLI prompts the user to either continue the reboot process without entering the Aboot shell or to restore the flash drive to the factory default state.

```
Press Control-C now to enter Aboot shell
^CAboot password:
incorrect password
Aboot password:
incorrect password
Aboot password:
incorrect password
Type "fullrecover" and press Enter to revert /mnt/flash to factory default
state, or just press Enter to reboot: fullrecover
All data on /mnt/flash will be erased; type "yes" and press Enter to
proceed,
or just press Enter to cancel:
```

The **fullrecover** operation replaces the flash contents with a factory default configuration. The CLI displays text similar to the following when performing a fullrecover, finishing with another entry option into the Aboot shell.

```
Erasing /mnt/flash
Writing recovery data to /mnt/flash
boot-config
startup-config
EOS.swi
210770 blocks
Restarting system.


Aboot 1.9.0-52504.EOS2.0


Press Control-C now to enter Aboot shell
```

6.4.3 File Structure

When you enter the Aboot CLI, the current working directory is the root directory on the switch. Switch image and configuration files are at **/mnt/flash.** When exiting the Aboot shell, only the contents of **/mnt/flash** are preserved. The **/mnt** directory contains the file systems of storage devices. Aboot mounts the internal flash device at **/mnt/flash**.

When a USB flash drive is inserted in one of the flash ports, Aboot mounts its file system on **/mnt/usb1**. The file system is unmounted when the USB flash drive is removed from the port. Most USB drives contain an LED that flashes when the system is accessing it; do not remove the drive from the flash port until the LED stops flashing.

6.4.4 Booting From the Aboot Shell

Aboot attempts to boot the software image (SWI) configured in ***boot-config*** automatically if you take no action during the boot process. If the boot process fails for any reason, such as an incorrectly configured SWI, Aboot enters the shell, allowing you to correct the configuration or boot an SWI manually. The **boot** command loads and boots a SWI file.

The **boot** command syntax is

boot *SWI*

where *SWI* lists the location of the EOS image that the command loads. *SWI* settings include:

- *DEVICE:PATH* Loads the SWI file from the specified storage device. The default *DEVICE* value is **flash;** other values include **file** and **usb1.**
- */PATH* Loads the SWI file from the specified path in the switch directory.
- **http:**//*SERVER/PATH* Loads an SWI file from the HTTP server on the host server.
- **ftp:**//*SERVER/PATH* Loads an SWI file from the FTP server on the host server.
- **tftp:**//*SERVER/PATH* Loads an SWI file from the TFTP server on the host server
- **nfs:**//*SERVER/PATH* Mounts path's parent directory from host server, loads SWI file from the loaded directory.

The boot command accepts the same commands as the SWI variable in the ***boot-config*** file. See Section 6.2.1.2: boot-config Command Line Content for a list of boot command formats.

If SWI is not specified in ***boot-config***, or if booting the SWI results in an error condition (for example, an incorrect path or unavailable HTTP server), Aboot halts the boot process and drops into the shell.

Example

- To boot EOS.swi from internal flash, enter one of these commands on the Aboot command line:
  - **boot flash:EOS.swi**
  - **boot /mnt/flash/EOS.swi**.

6.4.5 Commands

To list the contents of the internal flash, enter **ls /mnt/flash** at the Aboot# prompt.

Example

```
Aboot# ls /mnt/flash
EOS.swi boot-config startup-config
```

Commonly used commands include:

- `ls` Prints a list of the files in the current working directory
- `cd` Changes the current working directory
- `cp` Copies a file
- `more` Prints the contents of a file one page at a time
- `vi` Edits a text file
- `boot` Boots a SWI (see SWI section for information on specifying a SWI)
- `swiinfo` Prints information about a SWI
- `recover` Recovers the factory-default configuration
- `reboot` Reboots the switch
- `udhcpc` Configures a network interface automatically via DHCP
- `ifconfig` Prints or alters network interface settings
- `wget` Downloads a file from an HTTP or FTP server

Many Aboot shell commands are provided by Busybox, an open-source implementation of UNIX utilities. Busybox command help is found at http://www.busybox.net/downloads/BusyBox.html. Aboot provides access to only a subset of the documented commands.

Aboot can access networks through the Ethernet management ports. Aboot provides network interfaces mgmt1 and mgmt2. These ports are unconfigured by default; you can configure management port settings using Aboot shell commands like **ifconfig** and **udhcpc**. When a management interface is configured, use **wget** to transfer files from an HTTP or FTP server, **tftp** to transfer files from a TFTP server, or **mount** to mount an NFS filesystem.

6.5 Aboot Configuration Commands

This section describes the Aboot configuration commands that a ***boot-config*** file can contain.

- SWI . . . . . Page 216
- CONSOLESPEED. . . . . Page 213
- PASSWORD (ABOOT). . . . . Page 215
- NET commands . . . . . Page 214

CONSOLESPEED

CONSOLESPEED specifies the console baud rate. To communicate with the switch, the connected terminal must match the specified rate. Baud rates are 1200, 2400, 4800, 9600, 19200, or 38400.

The default baud rate is 9600.

Syntax

CONSOLESPEED=*baud_rate*

Parameters

- *baud_rate* specifies the console speed. Values include **1200**, **2400**, **4800**, **9600**, **19200**, or **38400**

Examples

- These lines are CONSOLESPEED command examples

```
CONSOLESPEED=2400
CONSOLESPEED=19200
```

NET commands

NETDEV indicates the network interface that ***boot-config*** network settings configure. If ***boot-config*** does not contain a NETDEV setting, the booting process does not attempt to configure a network interface. Other NET commands specify settings that Aboot uses to configure the interface.

Syntax

```
NETDEV=interface
NETAUTO=auto_setting
NETIP=interface_address
NETMASK=interface_mask
NETGW=gateway_address
NETDOMAIN=domain_name
NETDNS=dns_address
```

Parameters

- *interface* the network interface. Settings include:
  - **NETDEV=mgmt1** management port 1.
  - **NETDEV=mgmt2** management port 2.
- *auto_setting* the configuration method. Settings include
  - **NETAUTO=dhcp** interface is configured through a DHCP server; other NET commands are ignored.
  - **NETAUTO** command is omitted interface is configured manually with other NET commands,
- *interface_address* interface IP address, in dotted-decimal notation.
- *interface_mask* interface subnet mask, in dotted-decimal notation.
- *gateway_address* default gateway IP address, in dotted decimal notation.
- *domain_name* interface domain name.
- *dns_address* IP address of the Domain Name Server, in dotted decimal notation.

Examples

- This NETDEV command specifies Ethernet management 1 port:

```
NETDEV=mgmt1
```

- This NETAUTO command configures the interface through a DHCP server, ignoring other NET settings:

```
NETAUTO=dhcp
```

- These NET commands that configures the interface manually:

```
NETIP=10.12.15.10
NETMASK=255.255.255.0
NETGW=10.12.15.24
NETDOMAIN=mycompany.com
NETDNS=10.12.15.13
```

PASSWORD (ABOOT)

PASSWORD specifies the Aboot password, as described in Section 6.4.2: Accessing the Aboot Shell. If ***boot-config*** does not contain a PASSWORD line, the Aboot shell does not require a password.

boot-config stores the password as an MD5-encrypted string as generated by the UNIX passwd program or the crypt library function from a clear text seed. When entering the Aboot password, the user types the clear text seed.

There is no method of recovering the password from the encrypted string. If the clear text password is lost, delete the corresponding PASSWORD command line from the ***boot-config*** file.

The EOS **boot secret** command is the recommended method of adding or modifying the PASSWORD configuration line.

Syntax

PASSWORD=*encrypted_string*

Parameters

- *encrypted_string* the encrypted string that corresponds to the clear-text Aboot password.

Example

- This line is a PASSWORD command example where the encrypted string corresponds with the clear text password **abcde**.

```
PASSWORD=$1$CdWp5wfe$pzNtE3ujBoFEL8vjcq7jo/
```

SWI

SWI specifies the location and file name of the EOS image file that Aboot loads when booting, using the same format as the boot command to designate a local or network path.

Syntax

SWI=*file_location*

Parameters

- *file_location* specifies the location of the EOS image file. Formats include:
  - *device:path* – storage device location:

    device denotes a storage device. Settings include **flash**, **file** and **usb1**. Default is **flash**.
    path denotes a file location.

    Examples

    ```
    SWI=flash:EOS.swi - flash drive location.
    SWI=usb1:/EOS1.swi - usb drive location.
    SWI=file:/tmp/EOSexp.swi - switch directory location.
    ```

  - */path* – switch directory location.

    Example

    ```
    SWI=/mnt/flash/EOS.swi
    ```

  - **http:**//*server/path* – HTTP server location.

    Example

    ```
    SWI=http://foo.com/images/EOS.swi
    ```

  - **ftp:**//*server/path* – FTP server location.

    Example

    ```
    SWI=ftp://foo.com/images/EOS.swi
    ```

  - **tftp:**//*server/path* – TFTP server location.

    Example

    ```
    SWI=tftp://foo.com/EOS.swi
    ```

  - **nfs:**//*server/path* – imports *path* from *server*, then mounts parent directory of the *path*

    Example

    ```
    SWI=nfs://foo.com/images/EOS.swi
    ```

6.6 Switch Booting Commands

This section contains descriptions of the CLI commands that this chapter references.

- boot console . . . . . Page 218
- boot secret . . . . . Page 219
- boot system . . . . . Page 221
- protocol . . . . . Page 222
- redundancy . . . . . Page 223
- redundancy force-switchover . . . . . Page 224
- reload . . . . . Page 225
- reload <scheduled> . . . . . Page 227
- show redundancy file-replication . . . . . Page 228
- show redundancy states . . . . . Page 229
- show redundancy switchover sso . . . . . Page 230
- show reload . . . . . Page 231
- show reload cause . . . . . Page 232

boot console

The **boot console** command configures terminal settings for serial devices connecting to the console port. Console settings that you can specify from the boot command include:

- speed

Factory-default console settings are 9600 baud, no parity, 8 character bits, and 1 stop bit. If you do not know the current settings, restore the factory-default settings as described in Section 2.3.3: Restoring the Factory Default EOS Image and Startup Configuration.

Command Mode

Global Configuration

Command Syntax

boot console speed *baud*

Parameters

- *baud* console baud rate. Settings include 1200, 2400, 4800, 9600, 19200, and 38400.

Examples

- This command sets the console speed to 4800 baud

```
main-host(config)#boot console speed 4800
```

This code displays the result of the command:

```
main-host(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: 4800
Aboot password (encrypted): (not set)
```

The previous command places this command in the boot-command file.

```
CONSOLESPEED=4800
```

boot secret

The **boot secret** command creates or edits the Aboot shell password and stores the encrypted string in the PASSWORD command line of the ***boot-config*** file.

The **no boot secret** command removes the Aboot password from the ***boot-config*** file. When the Aboot password does not exist, entering Aboot shell does not require a password.

Command Mode

Global Configuration

Command Syntax

boot secret [*encrypt_type*] *password*

Parameters

- *encrypt_type* indicates the encryption level of the password parameter. Settings include:
  - — <no parameter> the *password* is clear text.
  - — **0** the *password* is clear text. Equivalent to the <no parameter> case.
  - — **5** the *password* is an md5 encrypted string.
- *password* specifies the boot password.
  - — if *encrypt_type* specifies clear text, then *password* must be in clear text.
  - — if *encrypt_type* specifies an encrypted string, then *password* must be an encrypted string.

Examples

- These equivalent commands set the Aboot password to xr19v:

```
main-host(config)#boot secret xr19v

main-host(config)#boot secret 0 xr19v
```

  This CLI code displays the result:

```
main-host(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): $1$k9YHFW8D$cgM8DSN.e/yY0p3k3RUvk.
```

  The CLI command places this PASSWORD line in the boot-command file.

```
PASSWORD=$1$k9YHFW8D$cgM8DSN.e/yY0p3k3RUvk.
```

  The user must enter ***xr19v*** at the login prompt to access the Aboot shell.

- These commands set the Aboot password to xr123, then displays the resulting ***boot-config*** code. The encrypted string was previously generated with xr123 as the clear text seed.

```
main-host(config)#boot secret 5 $1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
main-host(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): $1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
```

  The CLI command places this PASSWORD line in the boot-command file.

```
PASSWORD=$1$QfbYkVWb$PIXG0udEquW0wOSiZBN3D/
```

  The user must enter xr123 at the login prompt to access the Aboot shell.

- This command removes the Aboot password.

```
main-host(config)#no boot secret
```

This code displays the result of the command:

```
main-host(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: (not set)
Aboot password (encrypted): (not set)
```

Accessing the Aboot shell does not require a password.

boot system

The **boot system** command specifies the location of the EOS software image that Aboot loads when the switch boots. The command can refer to files on flash or on a module in the USB flash port.

Command Mode

Global Configuration

Command Syntax

boot system *device file_path*

Parameters

- *device* specifies the location of the image file. Settings include
  - **file:** file is located in the switch file directory.
  - **flash:** file is located in flash memory.
  - **usb1:** file is located on a drive inserted in the USB flash port. Available if a drive is in the port.
- *file_path* specifies the path and name of the file.

Examples

- This command designates EOS1.swi, on USB flash memory, as the EOS software image load file.

```
main-host(config)#boot system usb1:EOS1.swi
```

  The CLI command places this command in the boot-command file.

```
SWI=usb1:/EOS1.swi
```

- This command designates EOS.swi, on the switch flash, as the EOS software image load file.

```
main-host(config)#boot system flash:EOS.swi
```

  The CLI command places this command in the boot-command file.

```
SWI=flash:/EOS.swi
```

protocol

The **protocol** command configures how the supervisors will handle switchover events. By default, the switch is set to route processor redundancy (RPR), which synchronizes ***startup-config*** files between the supervisor modules and partially boots the standby supervisor. The mode can also be set to simplex (manual switchover only) or to stateful switchover (SSO) which synchronizes both ***startup-config*** and ***running-config*** files between the supervisor modules and fully boots the standby module to speed the switchover process and minimize packet loss.

The **no protocol** and **default protocol** commands set the redundancy protocol to the default value (**rpr**) by removing the **protocol** command from ***running-config***.

Command Mode

Redundancy Configuration

Command Syntax

```
protocol protocol_name
no protocol
default protocol
```

Parameters

- *protocol_name* specifies the location of the image file. Settings include
  - **rpr** route processor redundancy protocol (the default).
  - **simplex** no redundancy. Switchover must be initiated manually.
  - **sso** stateful switchover.

Examples

- These commands enter redundancy configuration mode and set the redundancy protocol to stateful switchover.

```
switch(config)#redundancy
switch(config-redundancy)#protocol sso
switch(config-redundancy)#
```

redundancy

The **redundancy** command places the switch in redundancy configuration mode.

Command Mode

Global Configuration

Command Syntax

redundancy

Related Commands

redundancy force-switchover Manually initiates a switchover.

Examples

- These commands enter redundancy configuration mode and set the redundancy protocol to stateful switchover.

```
switch(config)#redundancy
switch(config-redundancy)#protocol sso
switch(config-redundancy)#
```

redundancy force-switchover

The **redundancy force-switchover** command immediately switches control of the switch to the standby supervisor. If the redundancy mode is set to simplex, this is the only way to transfer control.

Command Mode

Privileged EXEC

Command Syntax

```
redundancy force-switchover
```

Related Commands

redundancy Places the switch in redundancy configuration mode.

Examples

- This command forces a switchover to the standby supervisor. The switchover is executed immediately without further confirmation from the user.

```
switch#redundancy force-switchover
This supervisor will be restarted.
```

reload

The **reload** command power cycles the switch, then resets it under Aboot control. The hard reset clears the switch, including memory states and other hardware logic

- Fixed 1-RU systems: The power supply remains powered up through the reset. Power is removed from all other switch components for two to five seconds.
- Modular systems: The power supply on the active supervisor remains powered up through the reset. Power is removed from all other supervisor components for at least one second. After the supervisor becomes functional, it manages the power-cycling of all line cards.

Command Mode

Privileged EXEC

Command Syntax

reload [**power**] [***CONFIRMATION***]

Parameters

- **power** Parameter has no functional affect.
- ***CONFIRMATION*** specifies when the switch resets.
  - — <no parameter> the switch requires a confirmation before starting the reset.
  - — **now** the reset begins immediately without prompting the user to confirm the request.

Related Commands

reload <scheduled> Schedules a pending reload operation.
show reload cause Displays cause of most recent reload.

Example

- Begin the reboot process by typing the **reload** command:

```
main-host#reload
```

When the **reload** command is entered, the switch sends a message prompting the user to save the configuration if it contains unsaved modifications, then asks the user to confirm the reload request.

```
System configuration has been modified. Save? [yes/no/cancel/diff]:n
Proceed with reload? [confirm]
```

The switch responds with a series of messages, including a notification that a ***system is being rebooted*** message was broadcast to all open CLI instances. The reload pauses to provide an option for the user to enter Aboot shell; the Aboot shell supports commands that restores the state of the internal flash to factory defaults or creates customized default state.

```
Broadcast message from root@mainStopping sshd: [  OK  ]
SysRq : Remount R/O
Restarting system

Aboot 1.9.0-52504.EOS2.0

Press Control-C now to enter Aboot shell
```

No action is required to continue the reset process. The switch displays messages to indicate the completion of individual tasks. The reboot is complete when the CLI displays a login prompt.

```
Booting flash:/EOS.swi
Unpacking new kernel
Starting new kernel
Switching to rooWelcome to Arista Networks EOS 4.4.0
Mounting filesystems:  [  OK  ]
Entering non-interactive startup
Starting EOS initialization stage 1: [  OK  ]
ip6tables: Applying firewall rules: [  OK  ]
iptables: Applying firewall rules: [  OK  ]
iptables: Loading additional modules: nf_conntrack_tftp [  OK  ]
Starting system logger: [  OK  ]
Starting system message bus: [  OK  ]
Starting NorCal initialization: [  OK  ]
Starting EOS initialization stage 2: [  OK  ]
Starting ProcMgr: [  OK  ]
Completing EOS initialization: [  OK  ]
Starting Power On Self Test (POST): [  OK  ]
Generating SSH2 RSA host key: [  OK  ]
Starting isshd: [  OK  ]
Starting sshd: [  OK  ]
Starting xinetd: [  OK  ]
[  OK  ] crond: [  OK  ]

main-host login:
```

reload <scheduled>

The **reload <scheduled>** command configures the switch to reset at a specified time or after a specified interval. Refer to **reload** for details on the functional details of the reset operation.

The switch prompts to save the configuration and confirm the reload request. After the request is confirmed, the switch resumes normal operation until the reload initiates.

The **reload cancel**, **no reload**, and **default reload** commands cancel the pending reload operation.

Command Mode

Privileged EXEC

Command Syntax

```
reload [power] TIMEFRAME [COMMENT]
reload cancel
no reload
default reload
```

Parameters

- **power** Parameter has no functional affect.
- *TIMEFRAME* specifies when the switch resets.
  - **at** *absolute_time* specifies the time when a reset begins.
  - **in** *relative_time* specifies the period until the reset begins.

  absolute_time designates a point in time with 24-hour clock notation: ***hh*:*mm*** [***month date***]
  relative_time designates a time period: ***hhh*:*mm***

- *COMMENT* descriptive text for denoting the reset reason. This option has no functional effect on the reset operation.
  - **reason** *comment_string* text that describes the reset.

Related Commands

reload Initiate an immediate reload operation.
show reload Displays time and reason of any pending reload operation.

Examples

- This command schedules a switch reset to begin in twelve hours.

```
switch#reload in 12:00
System configuration has been modified. Save? [yes/no/cancel/diff]:y
Proceed with reload? [confirm]
Reload scheduled for Tue Mar 27 05:57:25 2012 ( in 11 hours 59 minutes )
```

- This command cancels a scheduled switch reset.

```
switch#no reload
Scheduled reload has been cancelled
switch#
```

show redundancy file-replication

The **show redundancy file-replication** command displays the status and last synchronization date of file replication between the supervisors on the switch.

Command Mode

Privileged EXEC

Command Syntax

```
show redundancy file-replication
```

Related Commands

show redundancy states Displays status and redundancy protocol of supervisors.
show redundancy switchover sso Displays stateful switchover information since last reload.

Example

- This command displays the current file replication status of the supervisors.

```
switch#show redundancy file-replication
0 files unsynchronized, 2 files synchronized, 0 files failed, 2 files total.

File                    Status         Last Synchronized
----------------------- -------------- -----------------------
file:persist/sys        Synchronized   25 days, 19:48:26 ago
flash:startup-config    Synchronized   25 days, 19:48:26 ago
switch#
```

show redundancy states

The **show redundancy states** command displays the current status (active or standby) and the configured redundancy protocol of both supervisors, as well as summary information about the latest switchover event.

Command Mode

Privileged EXEC

Command Syntax

```
show redundancy states
```

Related Commands

show redundancy file-replication	Displays status of file replication between supervisors.
show redundancy switchover sso	Displays stateful switchover information since last reload.

Example

- This command displays redundancy information for both supervisors and a summary of the latest switchover.

```
switch#show redundancy states
  my state = ACTIVE
peer state = STANDBY HOT
      Unit = Secondary
   Unit ID = 2

Redundancy Protocol (Operational) = Stateful Switchover
Redundancy Protocol (Configured) = Stateful Switchover
Communications = Up
Ready for switchover

  Last switchover time = 25 days, 19:51:34 ago
Last switchover reason = Other supervisor stopped sending heartbeats
```

show redundancy switchover sso

The **show redundancy switchover sso** command displays the number of stateful switchovers since the last reload and a log of the events in the latest stateful switchover.

Command Mode

Privileged EXEC

Command Syntax

```
show redundancy switchover sso
```

Related Commands

show redundancy file-replication Displays status of file replication between supervisors.
show redundancy states Displays status and redundancy protocol of supervisors.

Example

- This command displays stateful switchover information.

```
switch#show redundancy switchover sso
Total number of Stateful Switchover completed since reload: 4

Latest Stateful Switchover occured 29 days, 12:48:22 ago @ 2012-06-09 19:47:50
(completed)
  0.000000: switchover started
  0.000235: stage PCIEAcquired started
  0.000349:   event PCIEAcquired:__dummyInternal1__ completed
  0.000394:   event PCIEAcquired:PlxPcie-system started
  0.027738:   event PCIEAcquired:PlxPcie-system completed
  0.027829: stage PCIEAcquired is complete
  0.027935: stage DmaReady started
  0.028042:   event DmaReady:ForwardingAgent started
  0.079620:   event DmaReady:ForwardingAgent completed
  0.079699: stage DmaReady is complete
  0.079781: stage TimeCriticalServices started
  0.079887:   event TimeCriticalServices:__dummyInternal1__ completed
  0.079928:   event TimeCriticalServices:Stp started
  0.208035:   event TimeCriticalServices:Stp completed
  0.208120: stage TimeCriticalServices is complete
<-------OUTPUT OMITTED FROM EXAMPLE-------->
 39.675076: stage NonCriticalServices started
 39.675145:   event NonCriticalServices:__dummyInternal1__ completed
 39.675183: stage NonCriticalServices is complete
 39.675399: switchover is complete
```

show reload

The **show reload** command displays the time and reason of any pending reload operation. The **reload <scheduled>** command schedules a reload operation and can be used to cancel a pending reload.

Command Mode

EXEC

Command Syntax

show reload

Related Commands

reload <scheduled> Schedules a pending reload operation.
show reload cause Displays cause of most recent reload.

Example

- These commands schedule a reload for 2:45 pm, displays the time of the pending reload, then cancels the scheduled reload.

```
switch>reload at 14:45
Proceed with reload? [confirm]
Reload scheduled for Tue Mar 27 14:45:00 2012 ( in 4 hours 11 minutes )
switch#show reload
Reload scheduled for Tue Mar 27 14:45:00 2012 ( in 4 hours 11 minutes )
switch#reload cancel
Scheduled reload has been cancelled
switch>
```

show reload cause

The **show reload cause** command displays the reason of the most recent reload operation. The command displays recommended actions and debug information related to the executed reload.

Command Mode

EXEC

Command Syntax

```
show reload cause
```

Related Commands

reload Initiates an immediate reload operation.
show reload Displays time and reason of all pending reload operations.

Example

- This command displays the cause of the recent reload operation.

```
switch>show reload cause
Reload Cause 1:
-------------------
Reload requested by the user.

Recommended Action:
-------------------
No action necessary.

Debugging Information:
----------------------
None available.
switch>
```

Chapter 7

Switch Environment Control

The following sections describe the commands that display temperature, fan, and power supply status:

- Section 7.1: Environment Control Introduction
- Section 7.2: Environment Control Overview
- Section 7.3: Configuring and Viewing Environment Settings
- Section 7.4: Environment Commands

The switch chassis, fans, power supplies, linecards, and supervisors also provide LEDs that signal status and conditions that require attention. The Quick Start Guide for the individual switches provides information about their LEDs.

7.1 Environment Control Introduction

Arista Networks switching platforms are designed to work reliably in common data center environments. To ensure their reliable operation and to monitor or diagnose the switch's health, Arista provides a set of monitoring capabilities available through the CLI or SNMP entity MIBs to monitor and diagnose potential problems with the switching platform.

7.2 Environment Control Overview

7.2.1 Temperature

Arista switches include internal temperature sensors. The number and location of the sensors vary with each switch model. Each sensor is assigned temperature thresholds that denote alert and critical conditions. Temperatures that exceed the threshold trigger the following:

- **Alert Threshold**: All fans run at maximum speed and a warning message is logged.
- **Critical Threshold**: The component is shut down immediately and its Status LED flashes orange.

In modular systems, cards are shut down when their temperatures exceed the critical threshold. The switch is shut down if the temperature remains above the critical threshold for three minutes.

7.2.2 Fans

Arista switches include fan modules that maintain internal components at proper operating temperatures. The number and type of fans vary with switch chassis type:

- **Fixed configuration switches** contain hot-swappable independent fans. Fan models with different airflow directions are available. All fans within a switch must have the same airflow direction.
- **Modular switches** contain independent fans that circulate air from front-to-rear panel. Power supplies for modular switches also include fans that cool the power supply and supervisors.

The switch operates normally when one fan is not operating. Nonfunctioning modules should not be removed from the switch unless they are immediately replaced; adequate switch cooling requires the installation of all components, including a non-functional fan.

Two non-operational fans trigger an ***insufficient fan shutdown*** condition. Under normal operations, this condition initiates a switch power down procedure.

Fans are accessible from the rear panel.

7.2.3 Power

Arista switches contain power supplies which provide power to internal components.

- **Fixed configuration switches** contain two power supplies, providing 1+1 redundancy.
- **Modular switches** contain four power supplies, providing a minimum of 2+2 redundancy.

Power supply LED indicators are visible from the rear panel.

7.3 Configuring and Viewing Environment Settings

7.3.1 Overriding Automatic Shutdown

7.3.1.1 Overheating

The switch can be configured to continue operating during temperature shutdown conditions. Ignoring a temperature shutdown condition is strongly discouraged because operating at high temperatures can damage the switch and void the warranty.

Temperature shutdown condition actions are specified by the **environment overheat action** command. The switch displays this warning when configured to ignore shutdown temperature conditions.

```
Switch(config)#environment overheat action ignore
======================================================================
WARNING: Overriding the system shutdown behavior when the system
is overheating is unsupported and should only be done under
the direction of an Arista Networks engineer. You risk damaging
hardware by not shutting down the system in this situation, and doing
so without direction from Arista Networks can be grounds for voiding
your warranty. To re-enable the shutdown-on-overheat behavior, use
the 'environment overheat action shutdown' command.
======================================================================
Switch(config)#
```

The ***running-config*** contains the **environment overheat action** command when it is set to **ignore**. When the command is not in ***running-config***, the switch shuts down when an overheating condition exists.

The following ***running-config*** file lists the **environment overheat action** command.

```
Switch#show running-config
! device: main-host (DCS-7124S, EOS-4.4.0)
!
username david secret 5 $1$o0WIXyim$dbYM4M/s/ol6Ytas8WlvY/

<-------OUTPUT OMITTED FROM EXAMPLE-------->

ip route 0.0.0.0/0 10.255.255.1
!
environment overheat action ignore
!
!
end
Switch#
```

7.3.1.2 Insufficient Fans

The switch can be configured to ignore the ***insufficient fan shutdown*** condition. This is strongly discouraged because continued operation without sufficient cooling may lead to a critical temperature condition that can damage the switch and void the warranty.

Insufficient-fans shutdown override is configured by the **environment insufficient-fans action** command. The switch displays this warning when configured to ignore insufficient-fan conditions.

```
Switch(config)#environment insufficient-fans action ignore
======================================================================
WARNING: Overriding the system shutdown behavior when the system
has insufficient fans inserted is unsupported and should only be done under
the direction of an Arista Networks engineer. You risk damaging
hardware by not shutting down the system in this situation, and doing
so without direction from Arista Networks can be grounds for voiding
your warranty. To re-enable the shutdown-on-overheat behavior, use
the 'environment insufficient-fans action shutdown' command.
======================================================================
Switch(config)#
```

The ***running-config*** contains the **environment insufficient-fans action** command when it is set to ***ignore***. When ***running-config*** does not contain this command, the switch shuts down when it detects an insufficient-fans condition.

7.3.1.3 Fan Speed

The switch can be configured to override the automatic fan speed. The switch normally controls the fan speed to maintain optimal operating temperatures. The fans can be configured to operate at a constant speed regardless of the switch temperature conditions.

Fan speed override is configured by the **environment fan-speed** command. The switch displays this warning when its control of fan speed is overridden.

```
Switch(config)#environment fan-speed override 50
======================================================================
WARNING: Overriding the system fan speed is unsupported and should only
be done under the direction of an Arista Networks engineer.
You can risk damaging hardware by setting the fan speed too low
and doing so without direction from Arista Networks can be grounds
for voiding your warranty.
To set the fan speed back to automatic mode, use the
'environment fan-speed auto' command
======================================================================
Switch(config)#
```

The ***running-config*** contains the **environment fan-speed override** command if it is set to override. When ***running-config*** does not contain this command, the switch controls the fan speed.

7.3.2 Viewing Environment Status

7.3.2.1 Temperature Status

To display internal temperature sensor status, enter **show environment temperature**.

```
Switch>show environment temperature
System temperature status is: Ok
                                                                    Alert   Critical
Sensor  Description                          Temperature  Threshold  Threshold
------- ------------------------------------ ------------ ---------- ----------
1       Front-panel temp sensor                  22.000C        65C        75C
2       Fan controller 1 sensor                  23.000C        75C        85C
3       Fan controller 2 sensor                  28.000C        75C        85C
4       Switch chip 1 sensor                     40.000C       105C       115C
5       VRM 1 temp sensor                        48.000C       105C       110C
Switch>
```

System temperature status is the first line that the command that the command displays. **System temperature status** values indicate the following:

- ***Ok:*** All sensors report temperatures below the alert threshold.
- ***Overheating:*** At least one sensor reports a temperature above its alert threshold.
- ***Critical:*** At least one sensor reports a temperature above its critical threshold.
- ***Unknown:*** The switch is initializing.
- ***Sensor Failed:*** At least one sensor is not functioning.

7.3.2.2 Fans

The **show environment cooling** command displays the cooling and fan status.

Example

This command displays the fan and cooling status.

```
Switch>show environment cooling
System cooling status is: Ok
Ambient temperature: 22C
Airflow: front-to-back
Fan Tray  Status           Speed
--------- ---------------- ------
1         Ok                 35%
2         Ok                 35%
3         Ok                 35%
4         Ok                 35%
5         Ok                 35%
Switch>
```

7.3.2.3 Power

The **show environment power** command displays the status of the power supplies.

Example

This command displays the status of the power supplies:

```
Switch>show environment power
Power                                    Input    Output   Output
Supply  Model                Capacity    Current  Current  Power    Status
------- -------------------- --------- -------- -------- -------- -------------
1       PWR-650AC                 650W    0.44A   10.50A   124.0W Ok
Switch>
```

7.3.2.4 System Status

The **show environment all** command lists the temperature, cooling, fan, and power supply information that the individual show environment commands display, as described in Section 7.3.2.1, Section 7.3.2.2, and Section 7.3.2.3.

Example

This command displays the temperature, cooling, fan, and power supply status:

```
Switch>show environment all
System temperature status is: Ok
                                                                    Alert   Critical
Sensor  Description                          Temperature  Threshold  Threshold
------- ------------------------------------ ------------ ---------- ----------
1       Front-panel temp sensor                   22.750C        65C        75C
2       Fan controller 1 sensor                   24.000C        75C        85C
3       Fan controller 2 sensor                   29.000C        75C        85C
4       Switch chip 1 sensor                      41.000C       105C       115C
5       VRM 1 temp sensor                         49.000C       105C       110C

System cooling status is: Ok
Ambient temperature: 22C
Airflow: front-to-back
Fan Tray  Status           Speed
--------- --------------- ------
1         Ok                35%
2         Ok                35%
3         Ok                35%
4         Ok                35%
5         Ok                35%

Power                                    Input    Output   Output
Supply  Model                Capacity    Current  Current  Power    Status
------- -------------------- --------- -------- -------- -------- -------------
1       PWR-650AC                 650W    0.44A   10.50A   124.0W Ok
```

7.4 Environment Commands

This section contains descriptions of the CLI commands that this chapter references.

Environment Control Configuration Commands

- environment fan-speed . . . . . Page 240
- environment insufficient-fans action . . . . . Page 241
- environment overheat action . . . . . Page 242

Environment Display Commands

- show environment all . . . . . Page 243
- show environment cooling . . . . . Page 244
- show environment power . . . . . Page 245
- show environment temperature . . . . . Page 246

environment fan-speed

The **environment fan-speed** command determines the method of controlling the fan speed of the switch fans. The switch automatically controls the fan speed by default.

The switch normally controls the fan speed to maintain optimal operating temperatures. The fans can be configured to operate at a constant speed regardless of the switch temperature conditions.

Important Overriding the system fan speed is unsupported and should only be done under the direction of an Arista Networks engineer. You can risk damaging hardware by setting the fan speed too low. Doing so without direction from Arista Networks can be grounds for voiding your warranty.

Command Mode

Global Configuration

Command Syntax

```
environment fan-speed ACTION
```

Parameters

- *ACTION* fan speed control method. Valid settings include:
  - **auto** fan speed is controlled by the switch.

    This option restores the default setting by removing the **environment fan-speed override** command from the configuration.
  - **override** *percent* fan speed is set to the specified percentage of the maximum. Valid *percent* settings range from 30 to 100.

Examples

- This command overrides the automatic fan speed control and configures the fans to operate at 50% of maximum speed.

```
switch(config)#environment fan-speed override 50
================================================================
WARNING: Overriding the system fan speed is unsupported and should only
be done under the direction of an Arista Networks engineer.
You can risk damaging hardware by setting the fan speed too low
and doing so without direction from Arista Networks can be grounds
for voiding your warranty.
To set the fan speed back to automatic mode, use the
'environment fan-speed auto' command
================================================================
switch(config)#
```

- This command restores control of the fan speed to the switch.

```
switch(config)#environment fan-speed auto
switch(config)#
```

environment insufficient-fans action

The **environment insufficient-fans** command controls the switch response to the insufficient fan condition.

The switch operates normally when one fan is not operating. Nonfunctioning modules should not be removed from the switch unless they are immediately replaced; adequate switch cooling requires the installation of all components, including a non-functional fan.

Two non-operational fans trigger an ***insufficient fan shutdown*** condition. This condition normally initiates a power down procedure.

By default, the switch initiates a shutdown procedure when it senses an insufficient fan condition.

Important Overriding the system shutdown behavior when the system has insufficient fans inserted is unsupported and should only be done under the direction of an Arista Networks engineer. You risk damaging hardware by not shutting down the system in this situation, and doing so without direction from Arista Networks can be grounds for voiding your warranty.

Command Mode

Global Configuration

Command Syntax

```
environment insufficient-fans action REMEDY
```

Parameters

- *REMEDY* configures action when switch senses an insufficient fan condition. Settings include:
  - **ignore** switch continues operating when insufficient fans are operating.
  - **shutdown** switch shuts power down when insufficient fans are operating.

  The ***shutdown*** parameter restores default behavior by removing the **environment insufficient-fans** command from ***running-config***.

Examples

- This command configures the switch to continue operating after it senses an insufficient fan condition.

```
switch(config)#environment insufficient-fans action ignore
======================================================================
WARNING: Overriding the system shutdown behavior when the system
has insufficient fans inserted is unsupported and should only be done under
the direction of an Arista Networks engineer. You risk damaging
hardware by not shutting down the system in this situation, and doing
so without direction from Arista Networks can be grounds for voiding
your warranty. To re-enable the shutdown-on-overheat behavior, use
the 'environment insufficient-fans action shutdown' command.
======================================================================
```

- This command configures the switch to shut down when it senses an insufficient fan condition.

```
switch(config)#environment insufficient-fans action shutdown
switch(config)#
```

environment overheat action

The **environment overheat** command controls the switch response to an overheat condition. By default, the switch shuts down when it senses an overheat condition.

Important Overriding the system shutdown behavior when the system is overheating is unsupported and should only be done under the direction of an Arista Networks engineer. You risk damaging hardware by not shutting down the system in this situation, and doing so without direction from Arista Networks can be grounds for voiding your warranty.

Arista switches include internal temperature sensors. The number and location of the sensors vary with each switch model. Each sensor is assigned temperature thresholds that denote alert and critical conditions. Temperatures that exceed the threshold trigger the following:

- **Alert Threshold**: All fans run at maximum speed and a warning message is logged.
- **Critical Threshold**: The component is shut down immediately and its Status LED flashes orange.

In modular systems, cards are shut down when their temperatures exceed the critical threshold. The switch normally shuts down if the temperature remains above the critical threshold for three minutes.

Command Syntax

```
environment overheat action REMEDY
```

Parameters

- *REMEDY* reaction to an overheat condition. Default value is ***shutdown***.
  - **shutdown** switch shuts power down by an overheat condition.
  - **ignore** switch continues operating during an overheat condition.

Examples

- This command configures the switch to continue operating after it senses an overheat condition.

```
switch(config)#environment overheat action ignore
================================================================
WARNING: Overriding the system shutdown behavior when the system
is overheating is unsupported and should only be done under
the direction of an Arista Networks engineer. You risk damaging
hardware by not shutting down the system in this situation, and doing
so without direction from Arista Networks can be grounds for voiding
your warranty. To re-enable the shutdown-on-overheat behavior, use
the 'environment overheat action shutdown' command.
================================================================
switch(config)#
```

- This command configures the switch to shut down when it senses an insufficient fan condition.

```
switch(config)#environment overheat action shutdown
switch(config)#
```

show environment all

The **show environment all** command displays temperature, cooling, and power supply status.

Command Mode

EXEC

Command Syntax

```
show environment all
```

Examples

- This command displays the switch's temperature, cooling, and power supply status

```
switch>show environment all
System temperature status is: Ok
                                                                    Alert   Critical
Sensor  Description                          Temperature  Threshold  Threshold
------- ------------------------------------ ------------ ---------- ----------
1       Front-panel temp sensor                   31.000C        65C        75C
2       Fan controller 1 sensor                   32.000C        75C        85C
3       Fan controller 2 sensor                   38.000C        75C        85C
4       Switch chip 1 sensor                      50.000C       105C       115C
5       VRM 1 temp sensor                         60.000C       105C       110C

System cooling status is: Ok
Ambient temperature: 31C
Airflow: front-to-back
Fan Tray  Status           Speed
--------- --------------- ------
1         Ok                52%
2         Ok                52%
3         Ok                52%
4         Ok                52%
5         Ok                52%

Power                                    Input    Output   Output
Supply  Model                Capacity  Current  Current  Power    Status
------- -------------------- --------- -------- -------- -------- -------------
1       PWR-760AC                 760W    0.81A   11.00A   132.6W Ok
2       PWR-760AC                 760W    0.00A    0.00A     0.0W AC Loss

switch>
```

show environment cooling

The **show environment cooling** command displays fan status, air flow direction, and ambient temperature on the switch.

Command Mode

EXEC

Command Syntax

```
show environment cooling
```

Display Values

- **System cooling status**:
  - — ***Ok*** no more than one fan has failed or is not inserted.
  - — ***Insufficient fans*** more than one fan has failed or is not inserted. This status is also displayed if fans with different airflow directions are installed. The switch shuts down if the error is not resolved.
- **Ambient temperature** temperature of the surrounding area.
- **Airflow** indicates the direction of the installed fans:
  - — ***front-to-back*** all fans flow air from the front to the rear of the chassis.
  - — ***back-to-front*** all fans flow air from the rear to the front of the chassis.
  - — ***incompatible fans*** fans with different airflow directions are inserted.
  - — ***Unknown*** The switch is initializing.
- **Fan Tray Status** table displays the status and operating speed of each fan. Status values indicate the following conditions:
  - — ***OK*** The fan is operating normally.
  - — ***Failed*** The fan is not operating normally.
  - — ***Unknown*** The system is initializing.
  - — ***Not Inserted*** The system is unable to detect the specified fan.
  - — ***Unsupported*** The system detects a fan that the current software version does not support.

Example

- This command displays the fan status, air flow direction, and ambient switch temperature.

```
switch>show environment cooling
System cooling status is: Ok
Ambient temperature: 30C
Airflow: front-to-back
Fan Tray  Status           Speed
--------- --------------- ------
1         Ok                51%
2         Ok                51%
3         Ok                51%
4         Ok                51%
5         Ok                51%
switch>
```

show environment power

The **show environment power** command displays the status of all power supplies in the switch.

Command Mode

EXEC

Command Syntax

show environment power

Example

- This command displays the status of power supplies on the switch.

```
switch>show environment power
Power                                      Input    Output   Output
Supply  Model                Capacity  Current  Current  Power    Status
------- -------------------- --------- -------- -------- -------- -------------
1       PWR-760AC                 760W    0.81A   11.00A   132.8W Ok
2       PWR-760AC                 760W    0.00A    0.00A     0.0W AC Loss
switch>
```

show environment temperature

The **show environment temperature** command displays the operating temperature on the switch.

Command Mode

EXEC

Command Syntax

show environment temperature [*INFO_LEVEL*]

Parameters

- *INFO_LEVEL* specifies level of detail that the command displays. Options include:
  - <no parameter> displays table that lists the temperature and thresholds of each sensor.
  - **detail** displays data block for each sensor listing the current temperature and historic data.

Display Values

- **System temperature status** is the first line that the command displays. Values report the following:
  - ***Ok*** All sensors report temperatures below the alert threshold.
  - ***Overheating*** At least one sensor reports a temperature above its alert threshold.
  - ***Critical*** At least one sensor reports a temperature above its critical threshold.
  - ***Unknown*** The switch is initializing.
  - ***Sensor Failed*** At least one sensor is not functioning.

Examples

- This command displays a table that lists the temperature measured by each sensor.

```
switch>show environment temperature
System temperature status is: Ok
                                                                  Alert   Critical
Sensor  Description                          Temperature  Threshold  Threshold
------- ------------------------------------ ------------ ---------- ----------
1       Front-panel temp sensor                  30.750C        65C        75C
2       Fan controller 1 sensor                  32.000C        75C        85C
3       Fan controller 2 sensor                  38.000C        75C        85C
4       Switch chip 1 sensor                     50.000C       105C       115C
5       VRM 1 temp sensor                        60.000C       105C       110C
switch>
```

- This command lists the temperature listed by each sensor, and includes the number of previous alerts, the time of the last alert, and the time of the last temperature change.

```
switch>show environment temperature detail
TempSensor1 - Front-panel temp sensor
                                Current State      Count            Last Change
  Temperature                         30.750C
  Max Temperature                     35.000C             4 days, 23:35:24 ago
  Alert                                 False          0                  never

TempSensor2 - Fan controller 1 sensor
                                Current State      Count            Last Change
  Temperature                         32.000C
  Max Temperature                     36.000C             4 days, 23:32:46 ago
  Alert                                 False          0                  never

TempSensor3 - Fan controller 2 sensor
                                Current State      Count            Last Change
  Temperature                         38.000C
  Max Temperature                     41.000C             4 days, 23:37:56 ago
  Alert                                 False          0                  never

TempSensor4 - Switch chip 1 sensor
                                Current State      Count            Last Change
  Temperature                         51.000C
  Max Temperature                     53.000C             4 days, 23:35:16 ago
  Alert                                 False          0                  never

TempSensor5 - VRM 1 temp sensor
                                Current State      Count            Last Change
  Temperature                         60.000C
  Max Temperature                     62.000C             4 days, 22:54:51 ago
  Alert                                 False          0                  never

switch>
```

Chapter 8

Ethernet Ports

This chapter describes Ethernet ports supported by Arista switches. Sections covered in this chapter include:

- Section 8.1: Ethernet Ports Introduction
- Section 8.2: Ethernet Standards
- Section 8.3: Ethernet Physical Layer
- Section 8.4: Interfaces
- Section 8.5: Ethernet Configuration Procedures
- Section 8.6: Ethernet Configuration Commands

8.1 Ethernet Ports Introduction

Arista switches support a variety of Ethernet network interfaces: copper, fiber from 100M to 40Gb and in ranges from half a meter to over 40 km. This chapter describes the configuration and monitoring options available in Arista switching platforms.

8.2 Ethernet Standards

Ethernet, standardized in IEEE 802.3, is a family of communication technologies for local area networks. Devices communicating over Ethernet divide data streams into frames. Each frame contains addresses (source and destination), payload, and error checking cyclical redundancy check (CRC).

There are two optical fiber classifications: single-mode (SMF) and multi-mode (MMF).

- SMF is used for long distance communication. Light follows a single path through the fiber. SMF has a narrow core (8.3 μm), requiring a more precise termination and connection method.
- MMF is used for distances of less than 300 meters and have performance characteristics useful in data center networks. Light is routed through multiple paths, resulting in differential mode delay (DMD).

  MMF has a wider core (50 or 62.5 μm) and can be driven by low cost VCSEL lasers for short distances. MMF connectors are cheaper and easier to terminate reliably than SMF connectors. MMF ia also referred to as OM2 and OM3

8.2.1 10 Gigabit Ethernet

The 10 Gigabit Ethernet (10GbE) standard defines an Ethernet implementation with a nominal data rate of 10 billion bits per second. 10 gigabit Ethernet implements full duplex point to point links connected by network switches. Half duplex operation, hubs and CSMA/CD do not exist in 10GbE. The 10 gigabit Ethernet standard encompasses a number of different physical layer (PHY) standards. A networking device may support different PHY types through pluggable PHY modules. 10 gigabit Ethernet standards are named ***10GBASE-xyz***, as interpreted by Table 8-1.

x	y	z
media type or wavelength, if media type is fiber	PHY encoding type	Number of WWDM wavelengths or XAUI Lanes
C = Copper (twin axial) T = Twisted Pair S = Short (850 nm) L = Long (1310 nm) E = Extended (1550 nm) Z = Ultra extended (1550 nm)	R = LAN PHY (64B/66B) X = LAN PHY (8B/10B) W = WAN PHY(*) (64B/66B)	If omitted, value = 1 (serial) 4 = 4 WWDM wavelengths or XAUI Lanes

Table 8-1 10GBASE-xyz Interpretation

8.2.2 Gigabit Ethernet

The Gigabit Ethernet (GbE), defined by IEEE 802.3-2008, describes an Ethernet version with a nominal data rate of one billion bits per second. GbE cables and equipment are similar to those used in previous standards. While full-duplex links in switches is the typical implementation, the specification permits half-duplex links connected through hubs.

Gigabit Ethernet physical layer standards that Arista switches support include 1000BASE-X (optical fiber), 1000BASE-T (twisted pair cable), and 1000BASE-CX (balanced copper cable).

- 1000BASE-SX is a fiber optic standard that utilizes multi-mode fiber supporting 770 to 860 nm, near infrared (NIR) light wavelength to transmit data over distances ranging from 220 to 550 meters. 1000BASE-SX is typically used for intra-building links in large office buildings, co-location facilities and carrier neutral internet exchanges.
- 1000BASE-LX is a fiber standard that utilizes a long wavelength laser (1,270–1,355 nm), and a maximum RMS spectral width of 4 nm to transmit data up to 5 km. 1000BASE-LX can run on all common types of multi-mode fiber with a maximum segment length of 550 m.
- 1000BASE-T is a standard for gigabit Ethernet over copper wiring. Each 1000BASE-T network segment can be a maximum length of 100 meters.

8.2.2.1 10/100/1000 BASE-T

Arista switches provide 10/100/1000 BASE-T Mbps Ethernet out of band management ports. Auto-negotiation is enabled on these interfaces. Speed (10/100/1000), duplex (half/full), and flow control settings are available using the appropriate **speed forced** and **flowcontrol** commands.

8.3 Ethernet Physical Layer

The Ethernet physical layer (PHY) includes hardware components connecting a switch's MAC layer to the transceiver, cable, and ultimately a peer link partner. Data exist in digital form at the MAC layer. On the line side of the PHY, data exist as analog signals: light blips on optical fiber or voltage pulses on copper cable. Signals may be distorted while in transit and recovery may require signal processing.

Ethernet physical layer components include a PHY and a transceiver.

8.3.1 PHYs

The PHY provides translation services between the MAC layer and transceiver. It also assists to establish links between the local MAC layer and peer devices by detecting and signaling fault conditions. The PHY line-side interface receives Ethernet frames from the link partner as analog waveforms. The PHY uses signal processing to recover the encoded bits, then sends them to the MAC layer.

PHY line-side interface components and their functions include:

- Physical Medium Attachment (PMA): Framing, octet synchronization, scrambling / descrambling.
- Physical Medium Dependent (PMD): Consists of the transceiver.
- Physical Coding Sublayer (PCS): Performs auto-negotiation and coding (8B/10B or 64B/66B).

The MAC sublayer of the PHY provides a logical connection between the MAC layer and the peer device by initializing, controlling, and managing the connection with the peer.

Ethernet frames transmitted by the switch are received by the PHY system-side interface as a sequence of digital bits. The PHY encodes them into a media-specific waveform for transmission through the line-side interface and transceiver to the link peer. This encoding may include signal processing, such as signal pre-distortion and forward error correction.

PHY system-side interface components that their functions include:

- 10 Gigabit Attachment Unit Interface (XAUI): Connects an Ethernet MAC to a 10 G PHY.
- Serial Gigabit Media Independent Attachment (SGMII): Connects an Ethernet MAC to a 1G PHY.

8.3.2 Transceivers

A transceiver connects the PHY to an external cable (optical fiber or twisted-pair copper) and through a physical connector (LC jack for fiber or RJ-45 jack for copper).

- Optical transceivers convert the PHY signal into light pulses that are sent through optical fiber.
- Copper transceivers connect the PHY to twisted-pair copper cabling.

Arista Small Form-Factor Pluggable (SFP+) and Quad Small Form Factor Pluggable (QSFP+) modules and cables provide high-density, low-power Ethernet connectivity over fiber and copper media. Arista offers transceivers that span data rates, media types, and transmission distances.

Arista SFP+ and QSFP+ modules:

- 10GBASE-SR (Short Reach)
  - Multi-mode fiber
  - Link length maximum 300 meters
  - Optical interoperability with 10GBASE-SRL
- 10GBASE-SRL (Short Reach Lite)
  - Multi-mode fiber
  - Link length maximum 100 meters
  - Optical interoperability with 10GBASE-SR

- 10GBASE-LR (Long Reach)
  - Single-mode fiber
  - Link length maximum 10 km
- 10GBASE-LRM (Long Reach Multimode)
  - Multi-mode fiber (50 um and 62.5 um).
  - Link length maximum 220 meters
- 10GBASE-ER (Extended Reach)
  - Single-mode fiber
  - Link length maximum 40 km
- 10GBASE-DWDM (Dense Wavelength Division Multiplexing)
  - Single-mode fiber (43 color options)
  - Link length maximum 40 km
- 40GBASE-SR4 QSFP+
  - Parallel OM3 or 150m over OM4 MMF
  - Link length maximum 100 meters
- 1000BASE-SX (Short Haul)
  - Multi-mode fiber
  - Link length maximum 550 meter
- 1000BASE-LX (Long Haul)
  - Single-mode or multi-mode fiber
  - Link length maximum 10 km (single mode) or 550 meters (multi-mode)
- 1000BASE-T (RJ-45 Copper)
  - Category 5 cabling
  - Full duplex 1000Mbps connectivity

Arista Cabled SFP+ and QSFP+ modules:

- 10GBASE-CR SFP+ to 10GBASE-CR SFP+ Cables
  - Includes SFP+ connectors on both ends
  - Twinax copper cable
  - Link lengths of 0.5, 1, 2, 3, 5 and 7 meters
- 40GBASE-CR QSFP+ to 4 x 10GBASE-CR SFP+ twinax copper cables
  - Twinax copper cable
  - Link lengths of 0.5, 1, 2 and 3 meters
- 40GBASE-CR4 QSFP+ to QSFP+ twinax copper cables
  - Twinax copper cable
  - Link lengths of 1, 2, 3, 5 and 7 meters

Internal ports

Several Arista switches include internal ports that connect directly to an external cable through an RJ-45 jack. Internal ports available on Arista switches include:

- 10GBASE-T (7140T-8S, 7120T-4S)
- 100/1000BASE-T (7048T-A)
- 100/1000/10GBASE-T (7050-T, 7100-T)

8.4 Interfaces

Arista switches provide two physical interface types that receive, process, and transmit Ethernet frames: Ethernet interfaces and Management interfaces.

Each Ethernet interface is assigned a 48-bit MAC address and communicates with other interfaces by exchanging data packets. Each packet contains the MAC address of its source and destination interface. Ethernet interfaces establish link level connections by exchanging packets. Interfaces do not typically accept packets with a destination address of a different interface.

Ethernet data packets are frames. A frame begins with preamble and start fields, followed by an Ethernet header that includes source and destination MAC addresses. The middle section contains payload data, including headers for other protocols carried in the frame. The frame ends with a 32-bit cyclic redundancy check (CRC) field that interfaces use to detect data corrupted during transmission.

8.4.1 Ethernet Interfaces

Ethernet speed and duplex configuration options depend on the media type of the interface:

- 40GBASE-SR4 and 40GBASE-CR4: Default operation is as four 10G ports. Speed command options support their configuration as a single 40G port.
- 10GBASE-T: Ports autonegotiates speed, offering 10G and 1G full duplex. Preferred setting is 10G. Half duplex, 10M, and 100M are not supported.

  Available **speed forced** commands include 10GFull and 1GFull.
- 10GBASE (SFP+): Ports operate as 10G ports. **Speed** commands do not effect configuration.
- 1000BASE-T (Copper): Default setting is ***autonegotiate***, offering 1G full and 100M; preferred setting is 1G full. Autonegotiation that offers only 100M is available through **speed spf-1000baset auto** command. Half duplex and 10M are not supported.
- 1000BASE (fiber): Operates as 1 G full duplex port. Speed commands do not effect configuration.

8.4.2 Management Interfaces

The management interface is a layer 3 host port that is typically connected to a PC for performing out of band switch management tasks. Each switch has one or two management interfaces. Only one port is required to manage the switch. The second port, when available, provides redundancy.

Management interfaces are 10/100/1000 BASE-T interfaces. By default, auto-negotiation is enabled on management interfaces. All combinations of speed 10/100/1000 and full or half duplex is enforceable on these interfaces through **speed** commands.

Management ports are enabled by default. The switch cannot route packets between management ports and network (Ethernet interface) ports because they are in separate routing domains. When the PC is multiple hops from the management port, packet exchanges through layer 3 devices between the management port and PC may require the enabling of routing protocols.

The Ethernet management ports are accessed remotely over a common network or locally through a directly connected PC. An IP address and static route to the default gateway must be configured to access the switch through a remote connection.

8.5 Ethernet Configuration Procedures

8.5.1 Physical Interface Configuration Modes

The switch provides two configuration modes for modifying Ethernet parameters:

- Interface-Ethernet mode configures parameters for specified Ethernet interfaces.
- Interface-Management mode configures parameters for specified management Ethernet interfaces.

Physical interfaces cannot be not created or removed.

Multiple interfaces can be simultaneously configured. Commands are available for configuring Ethernet specific, layer 2, layer 3, and application layer parameters. Commands that modify protocol specific settings in Ethernet configuration mode are listed in the protocol chapters.

The **interface ethernet** command places the switch in Ethernet-interface configuration mode.

Example

- This command places the switch in Ethernet-interface configuration mode for Ethernet interfaces 5-7 and 10.

```
switch(config)#interface ethernet 5-7,10
switch(config-if-Et5-7,10)#
```

The **interface management** command places the switch in management configuration mode.

Example

- This command places the switch in management-interface configuration mode for management interface 1.

```
switch(config)#interface management 1
switch(config-if-Ma1)#
```

8.5.2 Assigning MAC Addresses to an Interface

Ethernet and Management interfaces are assigned a MAC address when manufactured. This default address is the ***burn in address***. The **mac-address** command assigns a MAC address to the configuration mode interface in place of the burn in address. The **no mac-address** command reverts the interface's current MAC address to its burn in address.

Examples

- This command assigns the MAC address of ***001c.2804.17e1*** to Ethernet interface 7.

```
switch(config-if-Et7)#mac-address 001c.2804.17e1
```

- This command displays the MAC address of Ethernet interface 7. The active MAC address is ***001c.2804.17e1***. The default address is ***001c.7312.02e2***.

```
switch(config-if-Et7)#show interface ethernet 7
Ethernet7 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.2804.17e1 (bia 001c.7312.02e2)
  Description: b.e45

<-------OUTPUT OMITTED FROM EXAMPLE-------->

switch(config-if-Et7)#
```

8.5.3 Referencing Modular Ports

Arista modular switches provide a maximum of 384 ports through installed linecards. The maximum number of linecards on a modular is eight (7508 switch) or four (7504 switch). Each linecard contains 48 ports which are controlled by six PetraA ASIC chips. Each chip controls eight ports.

Several CLI commands modify modular parameters for all ports on a specified linecard or controlled by a specified chip. This manual uses these conventions to reference modular components:

- *card_x* refers to a line card. *card_x* value ranges from 3 to 10 (7508 switch) or 3 to 6 (7504 switch).
- *chip_y* refers to a PetraA ASIC chip. *chip_y* value ranges from 0 to 5.
- *port_z* refers to a linecard port. *port_z* value ranges from 1 to 48.

The port set controlled by specified PetraA chips is identical on all linecards:

- chip 0 references ports 1 through 8
- chip 1 references ports 9 through 16
- chip 2 references ports 17 through 24
- chip 3 references ports 25 through 32
- chip 4 references ports 33 through 40
- chip 5 references ports 41 through 48

Commands that display Ethernet port status use the convention *card_x/port_z* to label the linecard-port location of modular ports:

Example

- This command displays the status of interfaces 1 to 10 on linecard4:

```
switch>show interface ethernet 4/1-10 status
Port      Name              Status       Vlan        Duplex  Speed Type
Et4/1                       connected    1            full    10G Not Present
Et4/2                       connected    1            full    10G Not Present
Et4/3                       connected    1            full    10G Not Present
Et4/4                       connected    1            full    10G Not Present
Et4/5                       connected    1            full    10G Not Present
Et4/6                       connected    1            full    10G Not Present
Et4/7                       connected    1            full    10G Not Present
Et4/8                       connected    1            full    10G Not Present
Et4/9                       connected    1            full    10G Not Present
Et4/10                      connected    1            full    10G Not Present
switch>
```

8.5.4 QSFP+ Modules

QSFP+ modules are supported on these Arista switches:

- DCS-7050S-64: interfaces 49–52 (four interfaces).
- DCS-7050T-64: interfaces 49–52 (four interfaces).
- DCS-7050Q-16: interfaces 1–16 (16 interfaces).

The following sections describe the configuration of QSFP+ ports.

8.5.4.1 QSFP+ Ethernet Interface Configuration

Each QSFP+ module Ethernet interface is configurable either as a single 40G port or as four 10G ports. The switch displays four ports for each interface. The status of the ports depends on the interface configuration:

- The /1 port is active (***connected*** or ***not connected***), regardless of the interface configuration.

- The /2, /3, and /4 ports are ***errdisabled*** when the interface is configured as a single 40 port;
- all ports are active (***connected*** or ***not connected***), when the interface is configured as four 10Gports.

Example

On DCS-7050S-64, interface 49 is a QSFP+ interface. Its ports are listed as 49/1, 49/2, 49/3, and 49/4. Port status depends on the interface configuration:

- 40G port configuration: 49/1 is ***connected*** or ***not connected***; 49/2, 49/3, and 49/4 is ***errdisabled***.
- 4x10G port configuration: 49/1, 49/2, 49/3, and 49/4 status is ***connected*** or ***not connected***.

The **speed forced 40gfull** command configures a QSFP+ Ethernet interface as a 40G port. The **no speed forced 40gfull** command configures a QSFP+ Ethernet interface as four 10G ports. These commands reset the forwarding agent, which disrupts traffic on all switch ports. These commands must be applied to the /1 port.

To configure a QSFP+ Ethernet interface as a single 40G port:

Step 1 Enter Interface Ethernet configuration mode for port /1 of the QSFP+ Ethernet interface.

```
switch(config)#interface ethernet 49/1
```

Step 2 Enter **speed force 40gfull** command:

```
switch(config-if-Et49/1)#speed forced 40gfull
```

This step restarts the forwarding agent, which disrupts traffic on all switch ports. The agent may require more than a minute to restart.

Step 3 Enter **show interface status** to confirm the change in configuration.

```
switch(config-if-Et49/1)#show interface status
Port      Name              Status       Vlan        Duplex  Speed Type
Et1                         connected    1           full    10G 10GBASE-SR
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et48                        connected    1           full    10G 10GBASE-SR
Et49/1                      connected    1           full    40G 40GBASE-CR
Et49/2                      errdisabled  1           full    10G 40GBASE-CR
Et49/3                      errdisabled  1           full    10G 40GBASE-CR
Et49/4                      errdisabled  1           full    10G 40GBASE-CR
Et50/1                      connected    1           full    10G 40GBASE-CR
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et49/1)#
```

To configure a QSFP+ Ethernet interface as a four 10G port interface:

Step 1 Enter Interface Ethernet mode for port /1 of the QSFP+ interface.

```
switch(config)#interface ethernet 49/1
```

Step 2 Enter **no speed force 40gfull** command.

```
switch(config-if-Et49/1)#no speed forced 40gfull
```

This step restarts the forwarding agent, which disrupts traffic on all switch ports. The agent may require more than a minute to restart.

Step 3 Enter **show interface status** to confirm the change in configuration.

```
switch(config-if-Et49/1)#show interface status
Port      Name              Status       Vlan         Duplex  Speed Type
Et1                       notconnect   1            full    10G Not Present
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
Et48                      connected    1            full    10G 10GBASE-SR
Et49/1                    connected    1            full    10G 40GBASE-CR
Et49/2                    connected    1            full    10G 40GBASE-CR
Et49/3                    connected    1            full    10G 40GBASE-CR
Et49/4                    connected    1            full    10G 40GBASE-CR
Et50/1                    connected    1            full    10G 40GBASE-CR
            <-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config-if-Et49/1)#
```

8.5.4.2 QSFP-SFP Interface Availability (DCS-7050Q-16)

The DCS-7050Q-16 contains the following interfaces:

- 16 QSFP+ interfaces: labeled 1-16
- 8 SFP+ interfaces: labeled 17-24

The switch supports the simultaneous operation of a maximum of 64 10G ports. This requires that one QSFP+ interface is disabled for every four SFP+ interfaces that are enabled. The switch enforces this limitation through two port groups, each containing one QSFP+ interface and a set of four SFP+ interfaces. In each port group, either the QSFP+ interface or the SFP+ interface set is enabled. The port groups are configured independent of each other.

- Port group 1 contains interface 15 (QSFP+) and interfaces 17-20 (SFP+).
- Port group 2 contains interface 16 (QSFP+) and interfaces 21-24 (SFP+).

Table 8-2 displays the port group configuration options.

Port Group 1	Port Group 2	QSFP Ports enabled	SFP Ports enabled	Default
QSFP+ enabled	QSFP+ enabled	16: Ports 1-16	none	Yes
QSFP+ enabled	SFP+ enabled	15: Ports 1-15	4: Ports 21-24	No
SFP+ enabled	QSFP+ enabled	15: Ports 1-14, 16	4: Ports 17-20	No
SFP+ enabled	SFP+ enabled	14: Ports 1-14	8: Ports 17-24	No

Table 8-2 Port Group Configuration Options

The **hardware port-group** command determines the interface configuration for the specified port group. This command restarts the forwarding agent, which disrupts traffic on all switch ports. The agent may require more than one minute to restart.

These commands enable the QSFP+ interfaces in both port groups:

```
switch(config)#hardware port-group 1 select Et15/1-4
switch(config)#hardware port-group 2 select Et16/1-4
```

These commands enable the SFP+ interfaces in both port groups:

```
switch(config)#hardware port-group 1 select Et17-20
switch(config)#hardware port-group 2 select Et21-24
```

Example

- These commands configure the switch to provide availability to 15 QSFP+ and four SFP+ interfaces by enabling the QSFP+ interface in port group 2 and the SFP+ interfaces in port group 1.

```
switch(config)#hardware port-group 1 select Et17-20
switch(config)#hardware port-group 2 select Et16/1-4
```

The **show hardware port-group** command displays the status of ports in the port groups.

Example

- This command displays the status of ports in the two port groups on a DCS-7050Q-16 switch.

```
switch>show hardware port-group

Portgroup: 1    Active Ports: Et15/1-4
Port            State
----------------------------------------
Ethernet17      ErrDisabled
Ethernet18      ErrDisabled
Ethernet19      ErrDisabled
Ethernet20      ErrDisabled
Ethernet15/1    Active
Ethernet15/2    Active
Ethernet15/3    Active
Ethernet15/4    Active

Portgroup: 2    Active Ports: Et16/1-4
Port            State
----------------------------------------
Ethernet16/1    Active
Ethernet16/2    Active
Ethernet16/3    Active
Ethernet16/4    Active
Ethernet21      ErrDisabled
Ethernet22      ErrDisabled
Ethernet23      ErrDisabled
Ethernet24      ErrDisabled
switch>
```

8.5.5 Autonegotiated Settings

Autonegotiation is the procedure by which two connected devices choose common transmission parameters, including speed, duplex setting, and flow control.

8.5.5.1 Speed and Duplex

The **speed** command configures the transmission speed and duplex setting for the configuration mode interface. The scope and effect of this command depends on the interface type:

- 10GBASE-T: Default is 10G full. **Speed** command affects interface.

  Default setting is ***autonegotiate***, offering 10G full, 1G full, and 100M full; preferred setting is 10G full. Half duplex and 10M are not supported. The interface accepts **speed forced** commands for the supported speed and duplex settings.

- 10GBASE (SFP+): Operates as 10G full port. **Speed** command does not affect interface.
- 1000BASE (copper): Default is ***autonegotiate***. **Speed** command affects interface.

Default setting is ***autonegotiate***, offering 1G full and 100M; preferred setting is 1G full. Autonegotiation that offers only 100M is available through **speed spf-1000baset auto** command. Half duplex and 10M are not supported. The interface accepts **speed forced** commands for the supported speed and duplex settings.

- 1000BASE (fiber): Operates as 1G full port. **Speed** command does not affect interface.
- 40GBASE (QSFP+): Default is 4x10G-full. **Speed forced 40gfull** affects interface.

  Default setting is as four 10G full duplex ports. **Speed forced 40gfull** configures interface as a single 40G full duplex port.
- 10/100/1000: Default is ***autonegotiate***. **Speed** command affects interface.

  Default setting is autonegotiate, offering 1G full, 100M full, 10M full, 1G half, 100M half, and 10M half; preferred setting is 1G full. The interface accepts **speed forced** commands for the supported speed and duplex options.

  Example
  - This command configures a 40GBASE interface as a 40G port.

```
switch(config-if-Et49/1)#speed forced 40gfull
```

8.5.5.2 Flow Control

Flow control is a data transmission option that temporarily stops a device from sending data because of a peer data overflow condition. A sending device may transmit data faster than the other end of the link can accept, resulting in an overflowing buffer. The receiving device sends a PAUSE frame, instructing the sending device to halt transmission for a specified period.

Flowcontrol commands configure administrative settings for flow control packets

- The **flowcontrol receive** command configures the port's ability to receive flow control pause frames.
  - *off*: port does not process pause frames that it receives.
  - *on*: port processes pause frames that it receives
  - *desired*: port autonegotiates; processes pause frames if peer is set to *send* or *desired*.
- The **flowcontrol send** command configures the port's ability to transmit flow control pause frames.
  - *off*: port does not send pause frames.
  - *on*: port sends pause frames.
  - *desired*: port autonegotiates; sends pause frames if peer is set to *receive* or *desired*.

Desired is not an available parameter option. Ethernet data ports cannot be set to ***desired***. Management ports are set to ***desired*** by default and with the **no flowcontrol receive** command.

The port linking process includes flow control negotiation. Ports must have compatible flow control settings to create a link. Compatible flow control settings include:

local port	peer port
receive on	send on or send desired
receive off	send off or send desired
receive desired	send on , send off, or send desired
send on	receive on or receive desired
send off	receive off or receive desired
send desired	receive on , receive off, or receive desired

Table 8-3 Compatible Settings for Flow Control Negotiation

Example

- These commands set the flow control receive and send to **on** on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#flowcontrol receive on
switch(config-if-Et5)#flowcontrol send on
switch(config-if-Et5)#
```

8.5.6 Displaying Ethernet Port Properties

Show commands are available to display various Ethernet configuration and operational status on each interface. Ethernet settings that are viewable include:

- Port Type
- PHY Status
- Negotiated Settings
- Flow Control
- Capabilities

Port Type

The port type is viewable from the output of **show interfaces status**, **show interfaces capabilities**, and **show interfaces transceiver properties** commands.

Examples

- This **show interfaces status** command displays the status of Ethernet interfaces 1-5.

```
switch>show interfaces status
Port      Name              Status       Vlan        Duplex  Speed Type
Et1                       connected    1           full    10G 10GBASE-SRL
Et2                       connected    1           full    10G 10GBASE-SRL
Et3                       connected    1           full    10G 10GBASE-SRL
Et4                       connected    1           full    10G 10GBASE-SRL
Et5                       notconnect   1           full    10G Not Present
switch>
```

- This **show interfaces capabilities** command displays the status of Ethernet interfaces 1 and 2.

```
switch>show interfaces ethernet 1-2 capabilities
Ethernet1
  Model:        DCS-7124S
  Type:         10GBASE-SRL
  Speed/Duplex: 10G/full
  Flowcontrol:  rx-(off,on),tx-(off,on)
Ethernet2
  Model:        DCS-7124S
  Type:         10GBASE-SRL
  Speed/Duplex: 10G/full
  Flowcontrol:  rx-(off,on),tx-(off,on)
switch>
```

- This command displays the media type, speed, and duplex properties for Ethernet interfaces 1.

```
switch>show interfaces ethernet 1 transceiver properties
Name : Et1
Administrative Speed: 10G
Administrative Duplex: full
Operational Speed: 10G (forced)
Operational Duplex: full (forced)
Media Type: 10GBASE-SRL
```

PHY

PHY information for each Ethernet interface is viewed by entering the **show interfaces phy** command.

Example

- This command summarizes PHY information for Ethernet interfaces 1-3.

```
switch>show interfaces ethernet 1-3 phy
Key:
   U    = Link up
   D    = Link down
   R    = RX Fault
   T    = TX Fault
   B    = High BER
   L    = No Block Lock
   A    = No XAUI Lane Alignment
   0123 = No XAUI lane sync in lane N

                                    State    Reset
Port           PHY state          Changes    Count PMA/PMD PCS   XAUI
-------------- --------------- -------- -------- ------- ----- --------
Ethernet1      linkUp             14518     1750 U..     U.... U.......
Ethernet2      linkUp             13944     1704 U..     U.... U.......
Ethernet3      detectingXcvr          3        1               D..A0123
switch>
```

Negotiated Settings

Speed, duplex, and flow control settings are displayed through the show interfaces capabilities, PHY information for each Ethernet interface is viewed by entering the **show interfaces capabilities**, **show flowcontrol**, and **show interfaces status**.

Examples

- This command displays speed/duplex and flow control settings for Ethernet interface 1.

```
switch>show interfaces ethernet 1 capabilities
Ethernet1
  Model:        DCS-7124S
  Type:         10GBASE-SRL
  Speed/Duplex: 10G/full
  Flowcontrol:  rx-(off,on),tx-(off,on)
switch>
```

- This command displays the flow control settings for Ethernet interfaces 1-2.

```
switch>show flowcontrol interface ethernet 1-2
Port       Send FlowControl  Receive FlowControl  RxPause       TxPause
           admin    oper     admin    oper
---------  -------- -------- -------- --------    ------------- -------------
Et1        off      off      off      off         0             0
Et2        off      off      off      off         0             0
switch>
```

- This command displays the speed type and duplex settings for Ethernet interfaces 1-2.

```
switch>show interfaces management 1-2 status
Port      Name              Status       Vlan        Duplex  Speed Type
Ma1                         connected    routed      a-full a-100M 10/100/1000
Ma2                         connected    routed      a-full   a-1G 10/100/1000
switch>
```

8.6 Ethernet Configuration Commands

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Commands – All Interfaces

- hardware port-group . . . . . Page 265
- interface ethernet . . . . . Page 267
- interface management . . . . . Page 268

Interface Configuration Commands – Ethernet and Management Interfaces

- flowcontrol receive . . . . . Page 263
- flowcontrol send . . . . . Page 264
- link-debounce . . . . . Page 269
- mac-address . . . . . Page 270
- speed . . . . . Page 286

Interface Display Commands

- show flowcontrol . . . . . Page 271
- show hardware port-group . . . . . Page 272
- show interfaces capabilities . . . . . Page 273
- show interfaces counters . . . . . Page 274
- show interfaces counters bins . . . . . Page 275
- show interfaces counters errors . . . . . Page 276
- show interfaces counters queue . . . . . Page 277
- show interfaces counters rates . . . . . Page 278
- show interfaces negotiation . . . . . Page 279
- show interfaces phy . . . . . Page 280
- show interfaces status . . . . . Page 282
- show interfaces status errdisabled . . . . . Page 283
- show interfaces transceiver . . . . . Page 284
- show interfaces transceiver properties . . . . . Page 285

flowcontrol receive

The **flowcontrol receive** command configures administrative settings for inbound flow control packets. Ethernet ports use flow control to delay packet transmission when port buffers run out of space. Ports transmit a pause frame when its buffer is full, signaling its peer port to delay sending packets for a specified period.

The **flowcontrol receive** command configures the port's ability to receive flow control pause frames.

- *off*: port does not process pause frames that it receives.
- *on*: port processes pause frames that it receives
- *desired*: port autonegotiates flow control; processes pause frames if the peer is set to ***send desired***.

  Desired is not an available parameter option. Ethernet data ports cannot be set to ***desired***. Management ports are set to ***desired*** by default and with the **no flowcontrol receive** command.

The port linking process includes flow control negotiation. Ports must have compatible flow control settings to create a link. Compatible flow control settings include:

local port	peer port
receive on	send on or send desired
receive off	send off or send desired
receive desired	send on , send off, or send desired

Table 8-4 Compatible Settings for Flow Control Negotiation

The **no flowcontrol receive** and **default flowcontrol receive** commands restore the default flow control setting for the configuration mode interface by removing the corresponding **flowcontrol receive** command from ***running-config***. The default setting is ***off*** for Ethernet data ports and ***desired*** for Management ports.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
flowcontrol receive STATE
no flowcontrol receive
default flowcontrol receive
```

Parameters

- *STATE* flow control receive setting. Options include
  - **on** Enables a local port to process pause frames that a remote port sends.
  - **off** Prevents a local port from processing pause frames.

Examples

- These commands set the flow control receive to **on** on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#flowcontrol receive on
switch(config-if-Et5)#
```

flowcontrol send

The **flowcontrol send** command configures administrative settings for outbound flow control packets. Ethernet ports use flow control to delay packet transmission when port buffers run out of space. Ports transmit a pause frame when its buffer is full, signaling its peer port to delay sending packets for a specified period.

The **flowcontrol send** command configures the port's ability to transmit flow control pause frames.

- *off*: port does not send pause frames.
- *on*: port sends pause frames.
- *desired*: port autonegotiates flow control; sends pause frames if the peer is set to ***receive desired***.

  Desired is not an available parameter option. Ethernet data ports cannot be set to ***desired***. Management ports are set to ***desired*** by default and with the **no flowcontrol send** command.

The port linking process includes flow control negotiation. Ports must have compatible flow control settings to create a link. Compatible flow control settings include:

local port	peer port
send on	receive on or receive desired
send off	receive off or receive desired
send desired	receive on , receive off, or receive desired

Table 8-5 Compatible Settings for Flow Control Negotiation

The **no flowcontrol send** and **default flowcontrol send** commands restore the default flow control setting for the configuration mode interface by removing the corresponding **flowcontrol send** command from ***running-config***. The default setting is ***off*** for Ethernet data ports and ***desired*** for Management ports.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
flowcontrol send STATE
no flowcontrol send
default flowcontrol send
```

Parameters

- *STATE* flow control send setting. Options include
  - **on** Enables a local port to send pause frames.
  - **off** Prevents a local port from sending pause frames.

Examples

- These commands set the flow control send to **on** on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#flowcontrol send on
switch(config-if-Et5)#
```

hardware port-group

The **hardware port-group** command configures a DCS-7050Q-16 port group to activate a 40GBASE (QSFP+) interface or four 10GBASE (SFP+) interfaces, affecting QSFP+ and SFP+ availability on the switch.

The DCS-7050Q-16 contains the following interfaces:

- 16 QSFP+ interfaces: Labeled 1-16. Each configured as a 40G port or four 10G ports
- 8 SFP+ interfaces: labeled 17-24. Each configured as a 10G port.

The switch supports the simultaneous operation of 64 10G ports, requiring the disabling of one QSFP+ interface for every four enabled SFP+ interfaces. This limitation is enforced through two port groups, each containing one QSFP+ interface and a set of four SFP+ interfaces. In each port group, either the QSFP+ interface or the SFP+ interface set is enabled. The port groups are configured independently.

- Port group 1 contains interface 15 (QSFP+) and interfaces 17-20 (SFP+).
- Port group 2 contains interface 16 (QSFP+) and interfaces 21-24 (SFP+).

Table 8-6 displays the port group configuration options.

Port Group 1	Port Group 2	QSFP+ Ports enabled	SFP+ Ports enabled	Default
QSFP+ enabled	QSFP+ enabled	16: Ports 1-16	none	Yes
QSFP+ enabled	SFP+ enabled	15: Ports 1-15	4: Ports 21-24	No
SFP+ enabled	QSFP+ enabled	15: Ports 1-14, 16	4: Ports 17-20	No
SFP+ enabled	SFP+ enabled	14: Ports 1-14	8: Ports 17-24	No

Table 8-6 Port Group Configuration Options

The **no hardware port-group** and **default hardware port-group** commands restore a port group's default setting by removing the corresponding **hardware port-group** command from ***running-config***. The QSFP+ interface is active by default in each port group.

Command Mode

Global Configuration

Command Syntax

```
hardware port-group group_number select PORT_LIST
no hardware port-group group_number
default hardware port-group group_number
```

Parameters

- *group_number* label of the port group. Valid options are **1** and **2**.
- ***PORT_LIST*** ports activated by command. Options depend on *group_number* value.
  - **Et15/1-4** activates QSFP+ port on port group 1. Available when *group_number* is 1.
  - **Et16/1-4** activates QSFP+ port on port group 2. Available when *group_number* is 2.
  - **Et17-20** activates SFP+ ports on port group 1. Available when *group_number* is 1.
  - **Et21-23** activates SFP+ ports on port group 2. Available when *group_number* is 2.

Guidelines

The **hardware port-group** command is available on on DCS-7050Q-16 switches.

Examples

- These commands enable the QSFP+ interface in port group 1 and the SFP+ interfaces in port group 2.

```
switch(config)#hardware port-group 1 select Et15/1-4
switch(config)#hardware port-group 2 select Et21-24
switch(config)#
```

interface ethernet

The **interface ethernet** command places the switch in Ethernet-interface configuration mode for the specified interfaces. The command can specify a single interface or multiple interfaces.

Ethernet interfaces are physical interfaces and are not created or removed.

Interface management commands include:

- description
- exit
- load-interval
- mtu
- shutdown (Interfaces)

Ethernet management commands include:

- flowcontrol
- mac-address
- speed

Chapters describing supported protocols and other features list additional configuration commands available from Ethernet interface configuration mode.

Command Mode

Global Configuration

Command Syntax

interface ethernet *e_range*

Parameters

- *e_range* Numerical label of Ethernet interfaces to be configured.

  Formats include a number, number range, or comma-delimited list of numbers and ranges. Valid numbers depend on the Ethernet interfaces available on the switch.

Example

- This command enters interface configuration mode for Ethernet interfaces 1 and 2:

```
switch(config)#interface ethernet 1-2
switch(config-if-Et1-2)#
```

- This command enters interface configuration mode for Ethernet interface 1:

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#
```

interface management

The **interface management** command places the switch in management-interface configuration mode for the specified interfaces. The list can specify a single interface or multiple interfaces if the switch contains more than one management interface.

Management interfaces are physical interfaces and are not created or removed.

Interface management commands include:

- description
- exit
- load-interval
- mtu
- shutdown (Interfaces)

Ethernet management commands include:

- flowcontrol
- mac-address
- speed

Chapters describing supported protocols and other features list additional configuration commands available from management-interface configuration mode.

Command Mode

Global Configuration

Command Syntax

`interface management` *m_range*

Parameters

- *m_range* specifies management interfaces to be configured.

  Formats include a number, number range, or comma-delimited list of numbers and ranges. Number range depends on the management interfaces available on the switch.

Examples

- This command enters interface configuration mode for management interfaces 1 and 2.

```
switch(config)#interface management 1-2
switch(config-if-Ma1-2)#
```

- This command enters interface configuration mode for management interface 1:

```
switch(config)#interface management 1
switch(config-if-Ma1)#
```

link-debounce

The **link-debounce** command configures the link debounce time for the configuration mode interface. Link debounce time is the time that advertisements for new link states are delayed after the link state is established. By default, debounce time is set to zero, disabling link debounce.

Debounce times for link up and link down transitions can be independently configurable.

- Link-up debounce time: the delay before an interface advertises link down to link up transitions.
- Link-down debounce time: the delay before an interface advertises link up to link down transitions.

The **no link-debounce** and **default link-debounce** commands restore the default debounce setting for the configuration mode interface by removing the corresponding **link-debounce** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
link-debounce time WAIT_TIME
no link-debounce
default link-debounce
```

Parameters

- *WAIT_TIME* link debounce period (milliseconds). Options include
  - — *<0 - 30000>* One debounce value assigned as both link up and link down.
  - — *<0 - 30000> <0 - 30000>* Two debounce values: link up is first, link down is second.

  All debounce values range from 0 (disabled) to 30000 (30 seconds).

Examples

- These commands set the link up and link down debounce period to 10 seconds on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#link-debounce time 10000
switch(config-if-Et5)#
```

- These commands set the link up debounce to 10 seconds and the and link down debounce period to zero on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#link-debounce time 10000 0
switch(config-if-Et5)#
```

- These commands set the link up debounce to zero and the and link down debounce period to 12.5 seconds on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#link-debounce time 0 12500
switch(config-if-Et5)#
```

mac-address

The **mac-address** command assigns a MAC address to the configuration mode interface. An interface's default MAC address is its burn-in address.

The **no mac-address** command reverts the interface to its default MAC address by removing the corresponding **mac-address** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

mac-address *address*
no mac-address

Parameters

- *address* MAC address assigned to the interface. Format is dotted hex notation (H.H.H). Disallowed addresses are 0.0.0 and FFFF.FFFF.FFFF.

Example

- This command assigns the MAC address of 001c.2804.17e1 to Ethernet interface 7, then displays interface parameters, including the assigned address.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#mac-address 001c.2804.17e1
switch(config-if-Et7)#show interface ethernet 7
Ethernet3 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 001c.2804.17e1 (bia 001c.7312.02e2)
  Description: b.e45
  MTU 9212 bytes, BW 10000000 Kbit
  Full-duplex, 10Gb/s, auto negotiation: off
  Last clearing of "show interface" counters never
  5 seconds input rate 7.84 kbps (0.0% with framing), 10 packets/sec
  5 seconds output rate 270 kbps (0.0% with framing), 24 packets/sec
     1363799 packets input, 222736140 bytes
     Received 0 broadcasts, 290904 multicast
     0 runts, 0 giants
     0 input errors, 0 CRC, 0 alignment, 0 symbol
     0 PAUSE input
     2264927 packets output, 2348747214 bytes
     Sent 0 broadcasts, 28573 multicast
     0 output errors, 0 collisions
     0 late collision, 0 deferred
     0 PAUSE output
switch(config-if-Et7)#
```

show flowcontrol

The **show interfaces flowcontrol** command displays administrative and operational flow control data for the specified interfaces. Administrative data is the parameter settings stored in ***running-config*** for the specified interface; the switch uses these settings to negotiate flow control with the peer switch. Operational data is the resolved flow control setting that controls the port's behavior.

Command Mode

Privileged EXEC

Command Syntax

```
show flowcontrol [INTERFACE]
show [INTERFACE] flowcontrol
```

Parameters

- *INTERFACE* Interface type and number for which flow control data is displayed.
  - — <no parameter> all interfaces.
  - — **ethernet** *e_range* Ethernet interfaces in the specified range.
  - — **management** *m_range* Management interfaces in the specified range.

    Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command displays the flow control settings for Ethernet interfaces 1-10.

```
switch>show flowcontrol interface ethernet 1-10
Port       Send FlowControl  Receive FlowControl  RxPause       TxPause
           admin    oper     admin    oper
---------  -------- -------- -------- --------    ------------- -------------
Et1        off      off      off      off         0             0
Et2        off      off      off      off         0             0
Et3        off      off      off      off         0             0
Et4        off      off      off      off         0             0
Et5        off      off      off      off         0             0
Et6        off      off      off      off         0             0
Et7        off      off      off      off         0             0
Et8        off      off      off      off         0             0
Et9        off      off      off      off         0             0
Et10       off      off      off      off         0             0
switch>
```

show hardware port-group

The **show hardware port-group** command displays the status of DCS-7050Q-16 port-groups. Port groups contain one QSFP+ interface and a set of four SFP+ interfaces. In each port group, either the QSFP+ interface or the SFP+ interface set is enabled. The port groups are configured independent of each other.

- Port group 1 contains interface 15 (QSFP+) and interfaces 17-20 (SFP+).
- Port group 2 contains interface 16 (QSFP+) and interfaces 21-24 (SFP+).

Command Mode

EXEC

Command Syntax

```
show hardware port-group
```

Examples

- This command displays the status of ports in the two port groups on a DCS-7050Q-16 switch.

```
switch>show hardware port-group

Portgroup: 1    Active Ports: Et15/1-4
Port            State
----------------------------------------
Ethernet17      ErrDisabled
Ethernet18      ErrDisabled
Ethernet19      ErrDisabled
Ethernet20      ErrDisabled
Ethernet15/1    Active
Ethernet15/2    Active
Ethernet15/3    Active
Ethernet15/4    Active

Portgroup: 2    Active Ports: Et16/1-4
Port            State
----------------------------------------
Ethernet16/1    Active
Ethernet16/2    Active
Ethernet16/3    Active
Ethernet16/4    Active
Ethernet21      ErrDisabled
Ethernet22      ErrDisabled
Ethernet23      ErrDisabled
Ethernet24      ErrDisabled
switch>
```

show interfaces capabilities

The **show interfaces capabilities** command displays the model number, interface type, duplex mode, and flow control settings of the specified interfaces. The capabilities command is available on Ethernet and management interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [***INTERFACE***] **capabilities**

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.

  Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command displays the model number, interface type, duplex mode and flow control settings for Ethernet interfaces 1 and 2.

```
switch>show interfaces ethernet 1-2 capabilities
Ethernet1
  Model:        DCS-7124S
  Type:         10GBASE-SRL
  Speed/Duplex: 10G/full
  Flowcontrol:  rx-(off,on),tx-(off,on)
Ethernet2
  Model:        DCS-7124S
  Type:         10GBASE-SRL
  Speed/Duplex: 10G/full
  Flowcontrol:  rx-(off,on),tx-(off,on)
switch>
```

show interfaces counters

The **show interfaces counters** command displays packet and byte counters for the specified interfaces. Counters displayed by the command include:

- inbound bytes
- inbound unicast packets
- inbound multicast packets
- inbound broadcast packets
- outbound bytes
- outbound unicast packets
- outbound multicast packets
- outbound broadcast packets

Command Mode

EXEC

Command Syntax

show interfaces [***INTERFACE***] **counters**

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

Related Commands

- **show interfaces counters bins**
- **show interfaces counters errors**
- **show interfaces counters queue**
- **show interfaces counters rates**

Examples

- This command displays byte and packet counters for Ethernet interfaces 1 and 2.

```
switch>show interfaces ethernet 1-2 counters
Port                InOctets     InUcastPkts     InMcastPkts     InBcastPkts
Et1               99002845169        79116358           75557            2275
Et2               81289180585        76278345           86422              11

Port               OutOctets    OutUcastPkts    OutMcastPkts    OutBcastPkts
Et1                4347928323         6085482          356173            2276
Et2                4512762190         5791718          110498              15
switch>
```

show interfaces counters bins

The **show interfaces counters bins** command displays packet counters, categorized by packet length, for the specified interfaces. Packet length counters that the command displays include:

- 64 bytes
- 65-127 bytes
- 128-255 bytes
- 256-511 bytes
- 512-1023 bytes
- 1024-1522 bytes
- larger than 1522 bytes

Command Mode

EXEC

Command Syntax

```
show interfaces [INTERFACE] counters bins
```

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - — <no parameter> all interfaces.
  - — **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - — **management** *m_range* Management interface range specified by *m_range*.
  - — **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

Related Commands

- **show interfaces counters**
- **show interfaces counters errors**
- **show interfaces counters queue**
- **show interfaces counters rates**

Examples

- This command displays packet counter results for Ethernet interfaces 1 and 2.

```
switch>show interfaces ethernet 1-2 counters bins
Input
Port                64 Byte      65-127 Byte     128-255 Byte     256-511 Byte
-------------------------------------------------------------------------------
Et1                    2503         56681135          1045154          1029152
Et2                       8         50216275          1518179          1086297

Port          512-1023 Byte   1024-1522 Byte    1523-MAX Byte
-------------------------------------------------------------
Et1                  625825         17157823          8246822
Et2                  631173         27059077          5755101
switch>
```

show interfaces counters errors

The **show interfaces counters errors** command displays the error counters for the specified interfaces.

Command Mode
EXEC

Command Syntax
show interfaces [***INTERFACE***] **counters errors**

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

Display Values

The table displays the following counters for each listed interface:

- FCS: Inbound packets with CRC error and proper size.
- Align: Inbound packets with improper size (undersized or oversized).
- Symbol: Inbound packets with symbol error and proper size.
- Rx: Total inbound error packets.
- Runts: Outbound packets that terminated early or dropped because of underflow.
- Giants: Outbound packets that overflowed the receiver and were dropped.
- Tx: Total outbound error packets.

Related Commands

- **show interfaces counters**
- **show interfaces counters bins**
- **show interfaces counters queue**
- **show interfaces counters rates**

Examples

- This command displays the error packet counters on Ethernet interfaces 1-2.

```
switch>show interfaces ethernet 1-2 counters errors
Port               FCS    Align   Symbol       Rx    Runts   Giants       Tx
Et1                  0        0        0        0        0        0        0
Et2                  0        0        0        0        0        0        0
switch>
```

show interfaces counters queue

The **show interfaces counters queue** command displays the queue drop counters for the specified interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [***INTERFACE***] **counters queue**

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

Related Commands

- **show interfaces counters**
- **show interfaces counters bins**
- **show interfaces counters errors**
- **show interfaces counters rates**

Example

- This command displays the queue drop counters for Ethernet interfaces 1 and 2.

```
switch>show interfaces ethernet 1-2 counters queue
Port                  InDrops
Et1                       180
Et2                       169
switch>
```

show interfaces counters rates

The **show interfaces counters rates** command displays the received and transmitted packet rate counters for the specified interfaces. Counter rates provided include bytes (Mb/s), packets (kpacket/sec) and utilization percentage.

Command Mode

EXEC

Command Syntax

show interfaces [***INTERFACE***] **counters rates**

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

Related Commands

- **show interfaces counters**
- **show interfaces counters bins**
- **show interfaces counters errors**
- **show interfaces counters queue**

Example

- This command displays rate counters for Ethernet interfaces 1 and 2.

```
switch>show interfaces ethernet 1-2 counters rates
Port         Intvl   In Mbps      %  In Kpps  Out Mbps      % Out Kpps
Et1           0:05      53.3   0.5%        5      31.2   0.3%         2
Et2           0:05      43.3   0.4%        4       0.1   0.0%         0
switch>
```

show interfaces negotiation

The **show interfaces negotiation** command displays the speed, duplex, and flow control auto-negotiation status for the specified interfaces.

Command Mode
EXEC

Command Syntax
show interfaces [***INTERFACE***] **negotiation** [***INFO_LEVEL***]

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - — <no parameter> Display information for all interfaces.
  - — **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - — **management** *m_range* Management interface range specified by *m_range*.

  Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- *INFO_LEVEL* amount of information that is displayed. Options include:
  - — <no parameter> displays status and negotiated setting of local ports.
  - — **detail** displays status and negotiated settings of local ports and their peers.

Examples

- This command displays the negotiated status of management 1 and 2 interfaces

```
switch>show interface management 1-2 negotiation
Port       Autoneg            Negotiated Settings
           Status   Speed     Duplex    Rx Pause  Tx Pause
---------  -------  --------  --------  --------  --------
Ma1        success  100M      full      off       off
Ma2        success  auto      auto      off       off
switch>
```

- This command displays the negotiated status of management 1 interface and its peer interface.

```
switch>show interface management 1 negotiation detail
Management1 :

Auto-Negotiation Mode     10/100/1000 BASE-T (IEEE Clause 28)
Auto-Negotiation Status   Success

  Advertisements       Speed           Duplex     Pause
                       --------------- ---------- --------------------
      Local            10M/100M/1G     half/full  Disabled
      Link Partner     None            None       None

  Resolution           100Mb/s         full       Rx=off,Tx=off


switch>
```

show interfaces phy

The **show interfaces phy** command displays physical layer characteristics for the specified interfaces.

Command Mode
EXEC

Command Syntax
show interfaces [***INTERFACE***] **phy** [***INFO_LEVEL***]

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - — <no parameter> All interfaces.
  - — **ethernet** *e_range* Ethernet interfaces in specified range.

    Valid *e_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - — <no parameter> command displays table that summarizes phy data.
  - — **detail** command displays data block for each specified interface.

Examples

- This command summarizes PHY information for Ethernet interfaces 1-5.

```
switch>show interfaces ethernet 1-5 phy
Key:
   U    = Link up
   D    = Link down
   R    = RX Fault
   T    = TX Fault
   B    = High BER
   L    = No Block Lock
   A    = No XAUI Lane Alignment
   0123 = No XAUI lane sync in lane N

                                  State    Reset
Port            PHY state        Changes    Count PMA/PMD PCS   XAUI
--------------- --------------- -------- -------- ------- ----- --------
Ethernet1       linkUp             14518     1750 U..     U.... U.......
Ethernet2       linkUp             13944     1704 U..     U.... U.......
Ethernet3       linkUp             13994     1694 U..     U.... U.......
Ethernet4       linkUp             13721     1604 U..     U.... U.......
Ethernet5       detectingXcvr          3        1               D..A0123
switch>
```

- This command displays detailed phy information for Ethernet interface 1.

```
switch>show interfaces ethernet 1 phy detail
Current System Time: Mon Dec  5 11:32:57 2011
Ethernet1
                            Current State     Changes            Last Change
  PHY state                 linkUp              14523            0:02:01 ago
  HW resets                                      1751            0:02:07 ago
  Transceiver               10GBASE-SRL          1704            0:02:06 ago
  Transceiver SN            C743UCZUD
  Oper speed                10Gbps
  Interrupt Count                               71142
  Diags mode                normalOperation
  Model                     ael2005c
  Active uC image           microInit_mdio_SR_AEL2005C_28
  Loopback                  none
  PMA/PMD RX signal detect  ok                  11497            0:37:24 ago
  PMA/PMD RX link status    up                  11756            0:37:24 ago
  PMA/PMD RX fault          ok                  11756            0:37:24 ago
  PMA/PMD TX fault          ok                      0                  never
  PCS RX link status        up                   9859            0:02:03 ago
  PCS RX fault              ok                   9832            0:02:03 ago
  PCS TX fault              ok                    330            0:27:44 ago
  PCS block lock            ok                   9827            0:02:03 ago
  PCS high BER              ok                   8455            0:02:05 ago
  PCS err blocks            255                                  0:02:03 ago
  PCS BER                   16                  50092            0:02:05 ago
  XFI/XAUI TX link status   up                   1282            0:27:44 ago
  XFI/XAUI RX fault         ok                    585            0:27:44 ago
  XFI/XAUI TX fault         ok                   2142            0:02:05 ago
  XFI/XAUI alignment status ok                   2929            0:02:05 ago
  XAUI lane 0-3 sync        (0123) = 1111        2932            0:02:05 ago
  XAUI sync w/o align HWM   0                                          never
  XAUI sync w/o align max OK 5
  XAUI excess sync w/o align 0                                         never
  Xcvr EEPROM read timeout                         46    4 days, 6:33:45 ago
  Spurious xcvr detection                           0                  never
  DOM control/status fail                           0
  I2C snoop reset           0
  I2C snoop reset (xcvr)    0
  Margin count              5                last > 0            0:00:00 ago
  EDC resets                1                                    0:02:03 ago
  EDC FFE0 - FFE11          -4 -5 57 -6 -6 -2 1 0 -2 -1 1 -1
  EDC FBE1 - FBE4           6 -1 5 -1
  EDC TFBE1 - TFBE4         1 2 1 2
  EDC VGA1, VGA3            12 115
  TX path attenuation       3.0 dB
  TX preemphasis            (0,63,4) (pre,main,post)
switch>
```

show interfaces status

The **show interfaces status** command displays the interface name, link status, vlan, duplex, speed, and type of the specified interfaces. When the command includes a link status, the results are filtered to display only interfaces whose link status match the specified type.

Command Mode

EXEC

Command Syntax

show interfaces [***INTERFACE***] **status** [**connected**][**notconnect**][**disabled**]

Parameters (**connected, notconnect, disabled**) can be placed in any order.

Parameters

- ***INTERFACE*** Interface type and numbers. Options include:
  - <no parameter> All existing interfaces.
  - **ethernet** *e_range* Ethernet interfaces in the specified range.
  - **management** *m_range* Management interfaces in the specified range.
  - **port-channel** *p_range* All existing port-channel interfaces in the specified range.

  Valid *e_range*, *m_range*, and *p_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- ***STATUS_LEVEL*** interface status upon which the command filters output. Options include:
  - <no parameter> command does not filter on interface status.
  - **connected** interfaces connected to another port.
  - **notconnect** unconnected interfaces that are capable of connecting to another port
  - **disabled** interfaces that have been powered down or disabled.

Example

- This command displays the status of Ethernet interfaces 1-5.

```
switch>show interfaces status
Port      Name              Status       Vlan        Duplex  Speed Type
Et1                         connected    1            full    10G 10GBASE-SRL
Et2                         connected    1            full    10G 10GBASE-SRL
Et3                         connected    1            full    10G 10GBASE-SRL
Et4                         connected    1            full    10G 10GBASE-SRL
Et5                         notconnect   1            full    10G Not Present
switch>
```

show interfaces status errdisabled

The **show interfaces status errdisabled** command displays the specified interfaces that are in errdisabled state, including its link status, and the errdisable cause.

Command Mode
EXEC

Command Syntax
show interfaces [***INTERFACE***] **status errdisabled**

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.

  Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command displays the errdisabled ports.

```
switch>show interfaces status errdisabled
   Port         Name             Status         Reason
------------ ---------------- ----------------- ------------------
   Et49/2                      errdisabled    multi-lane-intf
   Et49/3                      errdisabled    multi-lane-intf
   Et49/4                      errdisabled    multi-lane-intf
switch>
```

show interfaces transceiver

The **show interfaces transceiver** command displays operational transceiver data for the specified interfaces.

Command Mode

EXEC

Command Syntax

show interfaces [***INTERFACE***] **transceiver** [***DATA_FORMAT***]

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.

  Valid *e_range*, and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- *DATA_FORMAT* format used to display the data. Options include:
  - <no parameter> table entries separated by tabs.
  - **csv** table entries separated by commas.

Related Commands

- **show interfaces transceiver properties**

Examples

- This command displays transceiver data on Ethernet interfaces 1 through 4.

```
switch>show interfaces ethernet 1-4 transceiver
If device is externally calibrated, only calibrated values are printed.
N/A: not applicable, Tx: transmit, Rx: receive.
mA: milliamperes, dBm: decibels (milliwatts).
                              Bias      Optical   Optical
        Temp       Voltage    Current   Tx Power  Rx Power  Last Update
Port    (Celsius)  (Volts)    (mA)      (dBm)     (dBm)     (Date Time)
-----   ---------  --------   --------  --------  --------  -------------------
Et1      34.17      3.30       6.75     -2.41     -2.83     2011-12-02 16:18:48
Et2      35.08      3.30       6.75     -2.23     -2.06     2011-12-02 16:18:42
Et3      36.72      3.30       7.20     -2.02     -2.14     2011-12-02 16:18:49
Et4      35.91      3.30       6.92     -2.20     -2.23     2011-12-02 16:18:45
switch>
```

show interfaces transceiver properties

The **show interfaces transceiver properties** command displays configuration information for the specified interfaces. Information provided by the command includes the media type, interface speed-duplex settings, speed-duplex operating state.

Command Mode

EXEC

Command Syntax

show interfaces [***INTERFACE***] **transceiver properties**

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **management** *m_range* Management interface range specified by *m_range*.

  Valid *e_range* and *m_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Related Commands

- **show interfaces transceiver**

Examples

- This command displays the media type, speed, and duplex properties for Ethernet interfaces 1-3.

```
switch>show interfaces ethernet 1-3 transceiver properties
Name : Et1
Administrative Speed: 10G
Administrative Duplex: full
Operational Speed: 10G (forced)
Operational Duplex: full (forced)
Media Type: 10GBASE-SRL

Name : Et2
Administrative Speed: 10G
Administrative Duplex: full
Operational Speed: 10G (forced)
Operational Duplex: full (forced)
Media Type: 10GBASE-SRL

Name : Et3
Administrative Speed: 10G
Administrative Duplex: full
Operational Speed: 10G (forced)
Operational Duplex: full (forced)
Media Type: 10GBASE-SRL

switch>
```

speed

The **speed** command configures the transmission speed and duplex setting for the configuration mode interface. The scope and effect of this command depends on the interface type. The **show interface status** command displays the interface type:

- 10GBASE-T: Default is 10G-full. **Speed** command affects interface.
- 10GBASE (SFP+): Default is 10G-full. **Speed** command does not affect interface.
- 1000BASE (copper): Default is 1G-full. **Speed spf-1000baset auto** affects interface.
- 1000BASE (fiber): Default is 1G-full. **Speed** command does not affect interface.
- 40GBASE (QSFP+): Default is 4x10G-full. **Speed forced 40gfull** configures interface as a 40G port.
- 10/100/1000: Default is ***auto-negotiation***. **Speed** command (10/100/1000 options) affects interface.

The **no speed** and **default speed** commands restore the default setting for the configuration mode interface by removing the corresponding **speed** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Management Configuration

Command Syntax

```
speed mode
no speed
default speed
```

Parameters

- *mode* transmission speed and duplex setting. Options include:
  - **auto** auto negotiation mode.
  - **sfp-1000baset auto** auto-negotiation mode (1000BASE-T interfaces only).
  - **forced 10000full** 10G full duplex.
  - **forced 1000full** 1G full duplex.
  - **forced 1000half** 1G half duplex.
  - **forced 100full** 100M full duplex.
  - **forced 100half** 100M half duplex.
  - **forced 10full** 10M full duplex.
  - **forced 10half** 10M half duplex.
  - **forced 40gfull** 40G full duplex.

    On 40GBASE (QSFP+) interfaces, the **forced 40gfull** and **no speed** options restart the forwarding agent, disrupting traffic on all ports for more than a minute.

Examples

- This command configures a 40GBASE interface as a 40G port.

```
switch(config)#interface ethernet 49/1
switch(config-if-Et49/1)#speed forced 40gfull
switch(config-if-Et49/1)#
```

- This command configures a 40GBASE interface as four 10G ports (default configuration).

```
switch(config-if-Et49/1)#no speed
switch(config-if-Et49/1)#
```

Chapter 9

Port Channels and LACP

This chapter describes channel groups, port channels, port channel interfaces, and the Link Aggregation Control Protocol (LACP). This chapter contains the following sections:

- Section 9.1: Port Channel Introduction
- Section 9.2: Port Channel Conceptual Overview
- Section 9.3: Configuration Procedures
- Section 9.4: Port Channel and LACP Configuration Commands

9.1 Port Channel Introduction

Arista's switching platforms support industry standard link aggregation protocols. Arista switches optimize traffic throughput by using MAC, IP addressing and services fields to effectively load share traffic across aggregated links. Managers can configure up to 16 ports into a logical port channel, either statically or dynamically through the IEEE Link Aggregation Control Protocol (LACP). Various negotiation modes are supported to accommodate any variety of configurations or peripheral requirements. There's even support for LACP fallback to support devices that need simple network connectivity to retrieve images or configurations prior to engaging port channel aggregation modes.

Arista's Multi-chassis Link Aggregation protocol (MLAG) (Chapter 12, starting on page 423) supports LAGs across paired Arista switches to provide both link aggregation and active/active redundancy. Up to 32 ports can be lagged through peered Arista switches to deliver over 320Gbps of bandwidth through a logical interface.

9.2 Port Channel Conceptual Overview

9.2.1 Channel Groups and Port Channels

A port channel is a communication link between two switches that consists of matching channel group interfaces on each switch. A port channel is also referred to as a Link Aggregation Group (LAG). Port channels combine the bandwidth of multiple Ethernet ports into a single logical link.

A channel group is a collection of Ethernet interfaces on a single switch. A port channel interface is a virtual interface that consists of a corresponding channel group and connects to a compatible interface on another switch to form a port channel. Port channel interfaces can be configured and used in a manner similar to Ethernet interfaces. Port channel interfaces are configurable as layer 2 interfaces, layer 3 (routable) interfaces, and VLAN members. Most Ethernet interface configuration options are available to port channel interfaces.

9.2.2 Link Aggregation Control Protocol (LACP)

The Link Aggregation Control Protocol (LACP), described by IEEE 802.3ad, defines a method for two switches to automatically establish and maintain LAGs. When LACP is enabled, a switch can configure a maximum of 16 LACP-compatible ports in a channel group. LACP terminology refers to the local interface as the ***actor*** and the remote interface as the ***partner***.

- In static mode, switches create port channels without awareness of their partner's port channels. Packets may drop when port channel static aggregate configurations differ between switches.

  The switch aggregates static links without LACP negotiation. The switches do not send LACP packets nor process inbound LACP packets.

- In dynamic mode, Link Aggregation Groups are aware of their partners' port channel states. Interfaces configured as dynamic LAGs are designated as ***active*** or ***passive***.

  — Active interfaces send LACP Protocol Data Units (LACP PDUs) at a rate of one per second when forming a channel with an interface on the peer switch. An aggregate forms if the peer runs LACP in active or passive mode.

  — Passive interfaces only send LACP PDUs in response to PDUs received from the partner. The partner switch must be in active mode and initiates negotiation by sending an LACP packet. The passive mode switch receives and responds to the packet to form a LAG.

An ***active*** interface can form port channels with ***passive*** or ***active*** partner interfaces. Port channels are not formed when the interface on each switch is ***passive***. Table 9-1 summarizes the valid LACP mode combinations:

Table 9-1 Valid LACP Mode Combinations

Switch 1	Switch 2	Comments
active	active	Links aggregate when LACP negotiation is successful.
active	passive	Links aggregate when LACP negotiation is successful.
passive	passive	Links aggregate without LACP.
on	—	Links aggregate without LACP.

During synchronization, interfaces transmit one LACP PDU per second. After synchronization is complete, interfaces exchange one PDU every thirty seconds, facilitated by a default timeout of 30 seconds and a failure tolerance of three. Under these parameters, when the switch does not receive an LACP PDU for an interface during a ninety second period, it records the partner interface as ***failed*** and removes the interface from the port channel.

Fallback mode allows an active LACP interface to establish a LAG before it receives PDUs from its peer. The fallback timer specifies the period the LAG remains active without receiving a peer PDU. Upon timer expiry, the interface reverts to static mode with one active port. An active interface that is not in fallback mode does not form a LAG until it receives PDUs from it peer.

The switch uses a link aggregation hash algorithm to determine the forwarding path within a Link Aggregation Group. The IP and MAC header fields can be selected as components of the hash algorithm.

9.3 Configuration Procedures

9.3.1 Configuring a Channel Group

Creating a Channel Group

The **channel-group** command assigns the configuration mode Ethernet interfaces to a channel group and specifies LACP attributes for the channel.

Channel groups are associated with a port channel interface immediately upon their creation. A command that creates a new channel group also creates a port channel with a matching ID. The port channel is configured in port-channel configuration mode. Configuration changes to a port channel interface propagate to all Ethernet interfaces in the corresponding channel group.

Example

These commands assign Ethernet interfaces 1 and 2 to channel group 10, enable LACP, and place the channel group in a negotiating state:

```
Switch(config)#interface ethernet 1-2
Switch(config-if-Et1-2)#channel-group 10 mode active
Switch(config-if-Et1-2)#
```

Adding an Interface to a Channel Group

The **channel-group** command adds the configuration mode interface to the specified channel group if the channel group exists. When adding channels to a previously created channel group, the LACP mode for the new channel must match the mode for the existing group.

Example

These commands add Ethernet interfaces 7 through 10 to previously created channel group 10, using the LACP trunking mode under which it was created.

```
Switch(config)#interface ethernet 7-10
Switch(config-if-Et7-10)#channel-group 10 mode active
Switch(config-if-Et7-10)#
```

Removing an Interface from a Channel Group

The **no channel-group** command removes the configuration mode interface from the specified channel group. Deleting all members of a channel group does not remove the associated port channel interface from ***running-config***.

Example

These commands remove add Ethernet interface 8 from previously created channel group 10.

```
Switch(config)#interface ethernet 8
Switch(config-if-Et8)#no channel-group
Switch(config-if-Et7-10)#
```

Deleting a Channel Group

A channel group is deleted by removing all Ethernet interfaces from the channel group. A channel group's LACP mode can be changed only be deleting the channel group and then creating an equivalent group with a different LACP mode.

Deleting a channel group by removing all Ethernet interfaces from the group preserves the port channel interface and its configuration settings.

View ***running-config*** to verify the deletion of all Ethernet interfaces from a channel group.

9.3.2 Configuring a Port Channel Interface

Creating a Port Channel Interface

The switch provides two methods for creating port channel interfaces:

- creating a channel group simultaneously creates an associated port channel.
- the **interface port-channel** command creates a port channel without assigning Ethernet channels to the new interface.

The **interface port-channel** command places the switch in interface-port channel configuration mode.

Example

This command creates port channel interface 8 and places the switch in port channel interface configuration mode:

```
Switch(config)#interface port-channel 8
Switch(config-if-Po8)#
```

Deleting a Port Channel Interface

The **no interface port-channel** command deletes the configuration mode port channel interface and removes the channel group assignment for each Ethernet channel assigned to the channel associated with the port channel. Removing all Ethernet interfaces from a channel group does not remove the associated port channel interface from ***running-config***.

9.3.3 Configuring LACP

Configuring the LACP Mode

The LACP mode is configured when a channel group is created. A channel group's LACP mode cannot be modified without deleting the entire channel group. A channel group's LACP mode can be altered without deleting the port channel interface associated with the channel group.

Example

These commands assign create a channel group and places it in LACP-active mode.

```
Switch(config)#interface ethernet 1-2
Switch(config-if-Et1-2)#channel-group 10 mode active
Switch(config-if-Et1-2)#
```

Configuring the System Priority

Each switch is assigned a globally unique system identifier by concatenating the system priority (16 bits) to the MAC address of one of its physical ports (48 bits). The system identifier is used by peer devices when forming an aggregation to verify that all links are from the same switch. The system identifier is also used when dynamically changing aggregation capabilities in response to LACP information; the system with the numerically lower system identifier is permitted to dynamically change advertised aggregation capabilities

The **lacp system-priority** command configures the switch's LACP system priority.

Example

This command assigns the system priority of 8192 to the switch.

```
Switch(config)#lacp system-priority 8192
Switch(config)#
```

Configuring Port Priority

LACP port priority determines the port that is active in a LAG in fallback mode. Numerically lower values have higher priority. Priority is supported on port channels with LACP-enabled physical interfaces.

The **lacp port-priority** command sets the aggregating port priority for the configuration mode interface.

Example

This command assigns the port priority of 4096 to Ethernet interface 1.

```
Switch(config-if-Et1)#lacp port-priority 4096
Switch(config-if-Et1)#
```

Configuring the LACP Packet Transmission Rate

The LACP transmission interval sets the rate at which LACP control packets are sent to an LACP-supported interface. Supported values include

- ***normal***: 30 seconds on synchronized interfaces; one second on interfaces that are synchronizing.
- ***fast***: one second.

The **lacp rate** command configures the LACP transmission interval on the configuration mode interface.

Example

This command sets the LACP rate to one second on Ethernet interface 4.

```
Switch(config-if-Et4)#lacp rate fast
Switch(config-if-Et4)#
```

Configuring LACP Fallback

LACP fallback mode allows an active LACP interface to establish a LAG before it receives PDUs from its peer. The **port-channel lacp fallback** command enables fallback mode on the configuration mode interface.

Example

This command enables LACP fallback mode on port-channel interface 13.

```
Switch(config-if-Po13)#port-channel lacp fallback
Switch(config-if-Po13)#
```

The **port-channel lacp fallback timeout** command specifies the period that a fallback-enabled interface can remain in LACP active mode without receiving an LACP PDU from its peer.

Example

This command configures an LACP fallback timeout period of 60 seconds.

```
Switch(config-if-Po13)#port-channel lacp fallback timeout 60
Switch(config-if-Po13)#
```

Configuring Minimum Links

The **port-channel min-links** command specifies the minimum number of interfaces that the configuration mode LAG requires to be active. This command is supported only on LACP ports. If there are fewer ports than specified by this command, the port channel interface does not become active.

Example

This command sets four as the minimum number of ports required by port channel 5 to be active.

```
switch(config-if-Po5)#port-channel min-links 4
switch(config-if-Po5)#
```

Load Balancing Hash Algorithms

The switch balances packet load across multiple links in a port channel by calculating a hash value based on packet header fields. The hash value determines the active member link through which the packet is transmitted. This method, in addition to balancing the load in the LAG, ensures that all packets in a data stream follow the same network path.

In network topologies that include MLAGs or multiple paths with equal cost (ECMP), programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links. This uneven distribution is avoided by performing different hash calculations on each switch routing the paths.

Hashing algorithm inputs depend on the ASIC hardware that controls switching functions. The following sections describe the hashing algorithms for each Arista hardware option.

- Hashing: FM4000 Hardware
- Hashing: Trident Hardware
- Hashing: petraA Hardware

The **port-channel load-balance fields** command specifies the hardware fields that configure the port channel load balance hash algorithm. The command description lists the hashing algorithms for each Arista hardware option.

Example

These commands configure an FM4000 switch's port channel load balance for IP packets by using the MAC destination and Ethernet type fields in the hashing algorithm.

```
Switch(config)#port-channel load-balance fm4000 fields ip mac-header
Switch(config)#port-channel load-balance fm4000 fields mac dst-mac eth-type
Switch(config)#
```

These commands perform the same function on a Trident platform switch.

```
Switch(config)#port-channel load-balance trident fields ip mac-header
Switch(config)#port-channel load-balance trident fields mac dst-mac eth-type
Switch(config)#
```

9.4 Port Channel and LACP Configuration Commands

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Commands

- interface port-channel . . . . . Page 296
- lacp system-priority . . . . . Page 299
- port-channel load-balance . . . . . Page 302
- port-channel load-balance fields . . . . . Page 303

Interface Configuration Commands – Ethernet Interface

- channel-group . . . . . Page 294
- lacp port-priority . . . . . Page 297
- lacp rate . . . . . Page 298

Interface Configuration Commands – Port Channel Interface

- port-channel lacp fallback . . . . . Page 300
- port-channel lacp fallback timeout . . . . . Page 301
- port-channel min-links . . . . . Page 307

EXEC Commands

- show etherchannel . . . . . Page 308
- show lacp aggregates . . . . . Page 309
- show lacp counters . . . . . Page 310
- show lacp interface . . . . . Page 311
- show lacp internal . . . . . Page 313
- show lacp neighbor . . . . . Page 314
- show lacp sys-id . . . . . Page 316
- show port-channel . . . . . Page 317
- show port-channel limits . . . . . Page 319
- show port-channel load-balance fields . . . . . Page 320
- show port-channel summary . . . . . Page 321
- show port-channel traffic . . . . . Page 322

channel-group

The **channel-group** command assigns the configuration mode Ethernet interfaces to a channel group and specifies LACP attributes for the channel. When adding channels to a previously created channel group, the LACP mode for the new channel must match the mode for the existing group.

Channel groups are associated with a port channel interface immediately upon their creation. A command that creates a new channel group also creates a port channel with a matching ID. The port channel is configured in port-channel configuration mode. Configuration changes to a port channel interface propagate to all Ethernet interfaces in the corresponding channel group. The **interface port-channel** command places the switch in interface-port-channel configuration mode.

The **no channel-group** and **default channel group** commands remove the configuration mode interface from the specified channel group.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
channel-group number LACP_MODE
no channel-group
default channel-group
```

Parameters

- *number* specifies a channel group ID. Values range from 1 through 1000.
- ***LACP_MODE*** specifies the interface LACP mode. Values include:
  - **mode on** Configures interface as a static port channel, disabling LACP. The switch does not verify or negotiate port channel membership with other switches.
  - **mode active** Enables LACP on the interface in active negotiating state. The port initiates negotiations with other ports by sending LACP packets.
  - **mode passive** Enables LACP on the interface in a passive negotiating state. The port responds to LACP packets but cannot start LACP negotiations.

Guidelines: Port Channels

You can configure a port channel to contain many ports, but only a subset may be active at a time. All active ports in a port channel must be compatible. Compatibility includes many factors and is platform specific. For example, compatibility may require identical operating parameters such as speed and maximum transmission unit (MTU). Compatibility may only be possible between specific ports because of the internal organization of the switch.

Guidelines: MLAG Configurations

Static LAG is not recommended in MLAG configurations. However, these considerations apply when the channel group mode is ***on*** while configuring static MLAG:

- When configuring multiple interfaces on the same static port channel:
  - all interfaces must physically connect to the same neighboring switch.
  - the neighboring switch must configure all interfaces into the same port channel.

  The switches are misconfigured when these conditions are not met.
- Disable the static port channel membership before moving any cables connected to these interfaces or changing a static port channel membership on the remote switch.

Example

- These commands assign Ethernet interfaces 8 and 9 to channel group 10, and enable LACP in negotiating mode.

```
switch(config)#interface ethernet 8-9
switch(config-if-Et8-9)#channel-group 10 mode active
switch(config-if-Et8-9)#show active
interface Ethernet8
   channel-group 10 mode active
interface Ethernet9
   channel-group 10 mode active
switch(config-if-Et8-9)#
```

interface port-channel

The **interface port-channel** command places the switch in port-channel interface configuration mode for modifying parameters of specified link aggregation (LAG) interfaces. When entering configuration mode to modify existing port channel interfaces, the command can specify multiple interfaces.

The command creates a port channel interface if the specified interface does not exist prior to issuing the command. When creating an interface, the command can only specify a single interface.

The **no interface port-channel** and **default interface port-channel** commands delete the specified LAG interfaces from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
interface port-channel p_range
no interface port-channel p_range
default interface port-channel p_range
```

Parameter

- *p_range* port channel interfaces (number, range, or comma-delimited list of numbers and ranges).

  VLAN number ranges from 1 to 1000.

Guidelines

When configuring a port channel, you do not first need to issue the **interface port-channel** command prior to assigning a port to the port channel (see the **channel-group** command). The port channel number is implicitly created when a port is added to the specified port channel with the **channel-group number** command.

To display ports that are members of a port channel, issue the **show port-channel** *number* command.

All active ports in a port channel must be compatible. Compatibility comprises many factors and is specific to a given platform. For example, compatibility may require identical operating parameters such as speed and/or maximum transmission unit (MTU). Compatibility may only be possible between specific ports because of internal organization of the switch.

To view information about hardware limitations for a port channel, issue the **show port-channel limits** command.

You can configure a port channel with a set of ports such that more than one subset of the member ports are mutually compatible. port channels in EOS are designed to activate the compatible subset of ports with the largest aggregate capacity. A subset with two 40 Gbps ports (aggregate capacity 80 Gbps) has preference to a subset with five active 10 Gbps ports (aggregate capacity 50 Gbps).

Example

- This example creates port channel interface 3:

```
switch(config)#interface port-channel 3
switch(config-if-Po3)#
```

lacp port-priority

The **lacp port-priority** command sets the aggregating port priority for the configuration mode interface. Priority is supported on port channels with LACP-enabled physical interfaces.

LACP port priority determines the port that is active in a LAG in fallback mode. Numerically lower values have higher priority. Priority is supported on port channels with LACP-enabled physical interfaces.

Each port in an aggregation is assigned a 32-bit port identifier by prepending the port priority (16 bits) to the port number (16 bits). Port priority determines the ports that are placed in standby mode when hardware limitations prevent a single aggregation of all compatible ports.

Priority numbers range from 0 to 65535. The default is 32768. Interfaces with higher priority numbers are placed in standby mode before interfaces with lower priority numbers.

The **no lacp port-priority** and **default lacp port-priority** commands restore the default port-priority to the configuration mode interface by removing the corresponding **lacp port-priority** command from *running-config*.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
lacp port-priority priority_value
no lacp port-priority
default lacp port-priority
```

Parameters

- *priority_level* port priority. Values range from 0 to 65535. Default is 32768

Example

- These commands assign the port priority of 4096 to Ethernet interface 8.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#lacp port-priority 4096
switch(config-if-Et8)#show active
interface Ethernet8
   lacp port-priority 4096
switch(config-if-Et8)#
```

lacp rate

The **lacp rate** command configures the LACP transmission interval on the configuration mode interface. The LACP timeout sets the rate at which LACP control packets are sent to an LACP-supported interface. Supported values include:

- ***normal***: 30 seconds with synchronized interfaces; one second while interfaces are synchronizing.
- ***fast***: one second.

This command is supported on LACP-enabled interfaces. The default value is ***normal***.

The **no lacp rate** and **default lacp rate** commands restore the default value of ***normal*** on the configuration mode interface by deleting the corresponding **lacp rate** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
lacp rate RATE_LEVEL
no lacp rate
default lacp rate
```

Parameters

- *RATE_LEVEL* LACP transmission interval . Options include:
  - **fast** one second.
  - **normal** 30 seconds for synchronized interfaces; one second while interfaces synchronize.

Examples

- This command sets the LACP rate to one second on Ethernet interface 4.

```
Switch(config-if-Et4)#lacp rate fast
Switch(config-if-Et4)#
```

lacp system-priority

The **lacp system-priority** command configures the switch's LACP system priority. Values range between 0 and 65535. Default value is 32768.

Each switch is assigned a globally unique 64-bit system identifier by prepending the system priority (16 bits) to the MAC address of one of its physical ports (48 bits). Peer devices use the system identifier when forming an aggregation to verify that all links are from the same switch. The system identifier is also used when dynamically changing aggregation capabilities resulting from LACP data; the system with the numerically lower system identifier can dynamically change advertised aggregation parameters.

The **no lacp system-priority and default lacp system-priority** commands restore the default system priority by removing the **lacp system-priority** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
lacp system-priority priority_value
no lacp system-priority
default lacp system-priority
```

Parameters

- *priority_value* system priority number. Values range from 0 to 65535. Default is 32768.

Example

- This command assigns the system priority of 8192 to the switch.

```
switch(config)#lacp system-priority 8192
switch(config)#
```

port-channel lacp fallback

The **port-channel lacp fallback** command enables LACP fallback mode on the configuration mode interface. Fallback mode allows an active LACP interface to establish a LAG before it receives PDUs from its peer. An active interface that is not in fallback mode does not form a LAG until it receives PDUs from it peer. The **port-channel lacp fallback timeout** command specifies the period the LAG remains active without receiving a peer PDU.

The **no port-channel lacp fallback** and **default port-channel lacp fallback** commands disable LACP fallback mode on the configuration mode interface by removing the corresponding **port-channel lacp fallback** command from ***running-config***.

LACP fallback is disabled by default.

Command Mode

Interface-Port-Channel Configuration

Command Syntax

```
port-channel lacp fallback
no port-channel lacp fallback
default port-channel lacp fallback
```

Example

- This command enables LACP fallback mode on port-channel interface 13.

```
switch(config)#interface port-channel 13
switch(config-if-Po13)#port-channel lacp fallback
switch(config-if-Po13)#show active
interface Port-Channel13
   port-channel lacp fallback
switch(config-if-Po13)#
```

port-channel lacp fallback timeout

The **port-channel lacp fallback timeout** command specifies the period a LAG in fallback mode remains active without receiving an LACP PDU from its peer. Upon timer expiry, the interface reverts to static mode with one active port. The default fallback timeout period is 90 seconds.

Command Mode

Interface-Port-Channel Configuration

Command Syntax

port-channel lacp fallback timeout *period*

Parameters

- *period* maximum interval between receipt of LACP PDU packets. Value ranges from 1 to 100 seconds. Default value is 90.

Example

- This command configures an LACP fallback timeout of 50 seconds on port channel interface 13.

```
switch(config)#interface port-channel 13
switch(config-if-Po13)#port-channel lacp fallback timeout 50
switch(config-if-Po13)#show active
interface Port-Channel13
   port-channel lacp fallback timeout 50
switch(config-if-Po13)#
```

port-channel load-balance

The **port-channel load-balance** command specifies the seed in the hashing algorithm that balances the load across ports comprising a port channel. Available seed values vary by switch platform.

The **no port-channel load-balance** and **default no port-channel load-balance** commands remove the command from ***running-config***, restoring the default hash seed value of 0.

Command Mode

Global Configuration

Command Syntax

```
port-channel load-balance HARDWARE number
no port-channel load-balance HARDWARE [number]
default port-channel load-balance HARDWARE [number]
```

Parameters

Parameter options vary by switch model. Verify available options with the CLI ? command.

- ***HARDWARE*** ASIC switching device. Value depends on the switch model:
  - **fm4000**
  - **trident**
- *number* The hash seed. Value range varies by switch platform.
  - when ***HARDWARE*=fm4000** number ranges from 0 to 2.
  - when ***HARDWARE*=trident** number ranges from 0 to 47.

    For trident platform switches, algorithms using hash seeds between 0 and 15 typically result in more effective distribution of data streams across the port channels.

Restrictions

This command is not available on Petra platform switches. The seed is set to zero on these switches.

Example

- This command configures the hash seed of 1 on FM4000 platform switches.

```
switch(config)#port-channel load-balance fm4000 1
switch(config)#
```

port-channel load-balance fields

The **port-channel load-balance fields** command specifies the hardware fields that configure the port channel load balance hash algorithm.

The switch calculates a hash value using the packet header fields to load balance packets across links in a port channel. The hash value determines the link through which the packet is transmitted. This method also ensures that all packets in a flow follow the same network path. Packet flow is modified by changing the inputs to the port channel hash algorithm.

In network topologies that include MLAGs, programming all switches to perform the same hash calculation increases the risk of hash polarization, which leads to uneven load distribution among LAG and MLAG member links in MLAG switches. This problem is avoided by performing different hash calculations between the MLAG switch, and a non-peer switch connected to it.

The hashing algorithm fields used for balancing IP packets differ from the fields used for non-IP packets.

Hashing algorithm inputs depend on the ASIC hardware that controls switching functions. The following sections describe the hashing algorithms for each Arista hardware option. Only one option is available per switch. Verify available options with the **CLI ?** command.

The **port-channel load-balance** command configures the hash seed for the algorithm.

Command Mode

Global Configuration

The following sections describe command options for each Arista hardware platform:

- Hashing: FM4000 Hardware
- Hashing: Trident Hardware
- Hashing: petraA Hardware

Hashing: FM4000 Hardware

Two load balancing commands configure the port channel hash:

- **port-channel load-balance fm4000 fields mac:** specifies the algorithm's use of MAC header fields. Available options include the MAC source address, MAC destination address, and Ethernet type. A command can use any combination of the options. The default setting is the selection of all options.
- **port-channel load-balance fm4000 fields ip**: specifies the algorithm's use of IP and MAC header fields. When ***ip-tcp-udp-header*** is selected, the algorithm uses source and destination IP addresses along with source and destination ports. When the ***mac-header*** is selected, the algorithm includes fields specified by the **port-channel load-balance fm4000 fields mac** parameter. A command must specify at least one option and may specify both. The default setting is the selection of both options.

The **port-channel load-balance fm4000 fields ip** command controls the hash algorithm for IP packets. The **port-channel load-balance fm4000 fields mac** command controls the hash algorithm for non-IP packets and affects the hash of IP packets if the IP command includes the ***mac- header***.

The **no port-channel load-balance fm4000 fields** and **default port-channel load-balance fm4000 fields** commands restore the default load distribution method by removing the corresponding **port-channel load-balance fm4000 fields** command from the configuration.

Command Syntax

```
port-channel load-balance fm4000 fields ip [IP__FIELD_NAME]
port-channel load-balance fm4000 fields mac [MAC_FIELD_NAME]
no port-channel load-balance fm4000 fields ip
no port-channel load-balance fm4000 fields mac
default port-channel load-balance fm4000 fields ip
default port-channel load-balance fm4000 fields mac
```

Parameters

- *IP_FIELD_NAME* fields the hashing algorithm uses for layer 3 routing. Options include:
  - **ip-tcp-udp-header**
  - **mac-header**
  - **ip-tcp-udp-header mac-header** options may be listed in any order
- *MAC_FIELD_NAME* fields the hashing algorithm uses for layer 2 routing. Options include
  - **dst-mac**
  - **eth-type**
  - **src-mac**
  - **dst-mac eth-type** options may be listed in any order
  - **dst-mac src-mac** options may be listed in any order
  - **eth-type src-mac** options may be listed in any order
  - **dst-mac eth-type src-mac** options may be listed in any order

Example

- These commands configure the switch's port channel load balance for IP packets by using the MAC destination and Ethernet type fields in the hashing algorithm.

```
switch(config)#port-channel load-balance fm4000 fields ip mac-header
switch(config)#port-channel load-balance fm4000 fields mac dst-mac eth-type
switch(config)#
```

Hashing: Trident Hardware

Two load balancing commands configure the port channel hash:

- **port-channel load-balance trident fields mac:** specifies the algorithm's use of MAC header fields. Available options include the MAC source address, MAC destination address, and Ethernet type. A command can use any combination of the options. The default setting is the selection of all options.
- **port-channel load-balance trident fields ip**: specifies the algorithm's use of IP and MAC header fields. When ***ip-tcp-udp-header*** is selected, the algorithm uses source and destination IP addresses along with source and destination ports. When the ***mac-header*** is selected, the algorithm includes fields specified by the **port-channel load-balance trident fields mac** parameter. A command must specify at least one option and may specify both. The default setting is the selection of both options.

The **port-channel load-balance trident fields ip** command controls the hash algorithm for IP packets. The **port-channel load-balance trident fields mac** command controls the hash algorithm for non-IP packets and affects the hash of IP packets if the IP command includes the ***mac- header***.

The **no port-channel load-balance trident fields** and **default port-channel load-balance trident fields** commands restore the default load distribution method by removing the corresponding **port-channel load-balance trident fields** command from the configuration.

Command Syntax

```
port-channel load-balance trident fields ip [IP__FIELD_NAME]
port-channel load-balance trident fields ipv6 [IP_FIELD_NAME]
port-channel load-balance trident fields mac [MAC_FIELD_NAME]
no port-channel load-balance trident fields ip
no port-channel load-balance trident fields ipv6
no port-channel load-balance trident fields mac
default port-channel load-balance trident fields ip
default port-channel load-balance trident fields ipv6
default port-channel load-balance trident fields mac
```

Parameters

- *IP_FIELD_NAME* fields the hashing algorithm uses for layer 3 routing. Options include:
  - **ip-tcp-udp-header**
  - **mac-header**
- *MAC_FIELD_NAME* fields the hashing algorithm uses for layer 2 routing. Options include:
  - **dst-mac**
  - **eth-type**
  - **src-mac**
  - **dst-mac eth-type** options may be listed in any order
  - **dst-mac src-mac** options may be listed in any order
  - **eth-type src-mac** options may be listed in any order
  - **dst-mac eth-type src-mac** options may be listed in any order

Example

- These commands configure the switch's port channel load balance for non IP packets by using the MAC destination and Ethernet type fields in the hashing algorithm.

```
switch(config)#port-channel load-balance trident fields mac dst-mac eth-type
switch(config)#
```

Hashing: petraA Hardware

One load balancing command configures the port channel hash:

- **port-channel load-balance petraA fields ip**: specifies the algorithm's use of IP and MAC header fields. When ***ip-tcp-udp-header***, the algorithm includes source and destination IP addresses along with, for TCP and UDP packets, source and destination ports. When **mac-header** is selected, the algorithm includes the entire MAC address header. A command can only specify one option. The default setting is ***ip-tcp-udp-header***.

The **port-channel load-balance petraA fields ip** command controls the port channel hash of IP packets. The port channel hash of non-IP packets always includes the entire MAC header.

The **no port-channel load-balance petraA fields ip** and **default port-channel load-balance petraA fields ip** commands restore the default load distribution method by removing the **port-channel load-balance fields ip** command from the configuration.

Command Syntax

```
port-channel load-balance petraA fields ip [IP__FIELD_NAME]
no port-channel load-balance petraA fields ip
default port-channel load-balance petraA fields ip
```

Parameters

- *IP_FIELD_NAME* fields the hashing algorithm uses for layer 3 routing. Options include:
  - **ip-tcp-udp-header**
  - **mac-header**

Example

- This command configures the switch's port channel load balance using IP packet fields.

```
switch(config)#port-channel load-balance petraA fields ip mac-header
switch(config)#
```

port-channel min-links

The **port-channel min-links** command specifies the minimum number of interfaces that the configuration mode LAG requires to be active. This command is supported only on LACP ports. If there are fewer ports than specified by this command, the port channel interface does not become active.

The default min-links value is 0.

Command Mode

Interface-Port-Channel Configuration

Command Syntax

port-channel min-links *quantity*

Parameters

- *quantity* minimum number of interfaces. Values range from 0 to 16. Default value is 0.

Example

- This command sets four as the minimum number of ports required by port channel 13 to be active.

```
switch(config)#interface port-channel 13
switch(config-if-Po13)#port-channel min-links 4
switch(config-if-Po13)#show active
interface Port-Channel13
   port-channel min-links 4
switch(config-if-Po13)#
```

show etherchannel

The **show port-channel** command displays information about members the specified port channels.

Command Mode

EXEC

Command Syntax

```
show port-channel [MEMBERS] [PORT_LIST] [INFO_LEVEL]
```

Parameters

- *MEMBERS* list of port channels for information is displayed. Options include:
  - <no parameter> all configured port channels.
  - *p_range* ports in specified channel list (number, number range, or list of numbers and ranges).
- *PORT_LEVEL* ports displayed, in terms of aggregation status. Options include:
  - <no parameter> Displays information on ports that are active members of the LAG.
  - **active-ports** Displays information on ports that are active members of the LAG.
  - **all-ports** Displays information on all ports (active or inactive) configured for LAG.
- *INFO_LEVEL* amount of information that is displayed. Options include:
  - <no parameter> Displays information at the brief level.
  - **brief** Displays information at the brief level.
  - **detail** Displays information at the detail level.

Display Values

- **Port Channel** Type and name of the port channel.
- **Time became active** Time when the port channel came up.
- **Protocol** Protocol operating on the port.
- **Mode** Status of the Ethernet interface on the port. The status value is Active or Inactive.
- **No active ports** Number of active ports on the port channel.
- **Configured but inactive ports** Ports configured but that are not actively up.
- **Reason unconfigured** Reason why the port is not part of the LAG.

Guidelines

The **show etherchannel** and **show port-channel** commands are identical. See **show port-channel** for additional information.

show lacp aggregates

The **show lacp aggregates** command displays aggregate IDs and the list of bundled ports for all specified port channels.

Command Mode
EXEC

Command Syntax
show lacp [***PORT_LIST***] **aggregates** [***PORT_LEVEL***] [***INFO_LEVEL***]

PORT_LEVEL and ***INFO_LEVEL*** parameters can be placed in any order.

Parameters

- ***PORT_LIST*** port channels for which aggregate information is displayed. Options include:
  - — <no parameter> all configured port channels.
  - — *c_range* channel list (number, range, or comma-delimited list of numbers and ranges).

  Port channel numbers range from 1 to 1000.
- ***PORT_LEVEL*** ports displayed, in terms of aggregation status. Options include:
  - — <no parameter> ports bundled by LACP into the port channel.
  - — **all-ports** all channel group ports, including channel group members not bundled into the port channel interface.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - — <no parameter> aggregate ID and bundled ports for each channel.
  - — **brief** aggregate ID and bundled ports for each channel.
  - — **detailed** aggregate ID and bundled ports for each channel.

Examples

- This command lists aggregate information for all configured port channels.

```
switch>show lacp aggregates
Port Channel Port-Channel1:
 Aggregate ID:
[(8000,00-1c-73-04-36-d7,0001,0000,0000),(8000,00-1c-73-09-a0-f3,0001,0000,0000)]
  Bundled Ports: Ethernet43 Ethernet44 Ethernet45 Ethernet46
Port Channel Port-Channel2:
 Aggregate ID:
[(8000,00-1c-73-01-02-1e,0002,0000,0000),(8000,00-1c-73-04-36-d7,0002,0000,0000)]
  Bundled Ports: Ethernet47 Ethernet48
Port Channel Port-Channel3:
 Aggregate ID:
[(8000,00-1c-73-04-36-d7,0003,0000,0000),(8000,00-1c-73-0c-02-7d,0001,0000,0000)]
  Bundled Ports: Ethernet3 Ethernet4
Port Channel Port-Channel4:
 Aggregate ID:
[(0001,00-22-b0-57-23-be,0031,0000,0000),(8000,00-1c-73-04-36-d7,0004,0000,0000)]
  Bundled Ports: Ethernet1 Ethernet2
Port Channel Port-Channel5:
 Aggregate ID:
[(0001,00-22-b0-5a-0c-51,0033,0000,0000),(8000,00-1c-73-04-36-d7,0005,0000,0000)]
  Bundled Ports: Ethernet41
switch>
```

show lacp counters

The **show lacp counters** command displays LACP traffic statistics.

Command Mode
EXEC

Command Syntax
show lacp [***PORT_LIST***] **counters** [***PORT_LEVEL***] [***INFO_LEVEL***]

PORT_LEVEL and ***INFO_LEVEL*** parameters can be placed in any order.

Parameters

- ***PORT_LIST*** ports for which port information is displayed. Options include:
  - <no parameter> all configured port channels
  - *c_range* ports in specified channel list (number, number range, or list of numbers and ranges).
  - **interface** ports on all interfaces.
  - **interface ethernet** *e_num* port on Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* loopback interface specified by *l_num*.
  - **interface management** *m_num* port on management interface specified by *m_num*.
  - **interface port-channel** *p_num* port on port channel interface specified by *p_num*.
  - **interface vlan** *v_num* port on VLAN interface specified by *v_num*.
  - **interface peerethernet***pe_num* port on peer Ethernet interface specified by p*e_num*.
  - **interface peerport-channel***pc_num* port on peer port channel interface specified by *pc_num*.
- ***PORT_LEVEL*** ports displayed, in terms of aggregation status. Options include:
  - <no parameter> only ports bundled by LACP into an aggregate.
  - **all-ports** all ports, including LACP candidates that are not bundled.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> displays packet transmission (TX and RX) statistics.
  - **brief** displays packet transmission (TX and RX) statistics.
  - **detailed** displays packet transmission (TX and RX) statistics and actor-partner statistics.

Examples

- This command displays transmission statistics for all configured port channels.

```
switch>show lacp counters brief
                      LACPDUs          Markers      Marker Response
Port   Status        RX          TX    RX     TX      RX         TX     Illegal
--------------------------------------------------------------------------------
Port Channel Port-Channel1:
Et43   Bundled   396979     396959    0      0       0          0           0
Et44   Bundled   396979     396959    0      0       0          0           0
Et45   Bundled   396979     396959    0      0       0          0           0
Et46   Bundled   396979     396959    0      0       0          0           0
Port Channel Port-Channel2:
Et47   Bundled   396836     396883    0      0       0          0           0
Et48   Bundled   396838     396883    0      0       0          0           0

switch>
```

show lacp interface

The **slow lacp interface** command displays port status for all port channels that include the specified interfaces. Within the displays for each listed port channel, the output displays sys-id, partner port, state, actor port, and port priority for each interface in the channel.

Command Mode

EXEC

Command Syntax

show lacp interface [***INTERFACE_PORT***] [***PORT_LEVEL***] [***INFO_LEVEL***]

INTERFACE_PORT is listed first when present. Other parameters can be listed in any order.

Parameters

- *INTERFACE_PORT* interfaces for which information is displayed. Options include:
  - <no parameter> all interfaces in channel groups.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* loopback interface specified by *l_num*.
  - **management** *m_num* management interface specified by *m_num*.
  - **port-channel** *p_num* port channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
  - **peerethernet***pe_num* peer Ethernet interface specified by *pe_num*.
  - **peerport-channel***pc_num* peer port-channel interface *pc_num*.
- *PORT_LEVEL* ports displayed, in terms of aggregation status. Options include:
  - <no parameter> command lists data for ports bundled by LACP into the aggregate.
  - **all-ports** command lists data for all ports, including LACP candidates that are not bundled.
- *INFO_LEVEL* amount of information that is displayed. Options include:
  - <no parameter> displays same information as **brief** option.
  - **brief** displays LACP configuration data, including sys-id, actor, priorities, and keys.
  - **detailed** includes **brief** option information plus state machine data.

Example

- This command displays LACP configuration information for all ethernet interfaces.

```
switch>show lacp interface
State: A = Active, P = Passive; S=ShortTimeout, L=LongTimeout;
       G = Aggregable, I = Individual; s+=InSync, s-=OutOfSync;
       C = Collecting, X = state machine expired,
       D = Distributing, d = default neighbor state
             |                       Partner                          Actor
Port Status  | Sys-id                 Port# State   OperKey PortPri  Port#
-------------------------------------------------------------------------------
Port Channel Port-Channel1:
Et43 Bundled | 8000,00-1c-73-09-a0-f3    43 ALGs+CD  0x0001   32768     43
Et44 Bundled | 8000,00-1c-73-09-a0-f3    44 ALGs+CD  0x0001   32768     44
Et45 Bundled | 8000,00-1c-73-09-a0-f3    45 ALGs+CD  0x0001   32768     45
Et46 Bundled | 8000,00-1c-73-09-a0-f3    46 ALGs+CD  0x0001   32768     46
Port Channel Port-Channel2:
Et47 Bundled | 8000,00-1c-73-01-02-1e    23 ALGs+CD  0x0002   32768     47
Et48 Bundled | 8000,00-1c-73-01-02-1e    24 ALGs+CD  0x0002   32768     48
```

```
              |                 Actor
Port Status   |   State         OperKey    PortPriority
--------------------------------------------------------
Port Channel Port-Channel1:
Et43 Bundled  |   ALGs+CD        0x0001           32768
Et44 Bundled  |   ALGs+CD        0x0001           32768
Et45 Bundled  |   ALGs+CD        0x0001           32768
Et46 Bundled  |   ALGs+CD        0x0001           32768
Port Channel Port-Channel2:
Et47 Bundled  |   ALGs+CD        0x0002           32768
Et48 Bundled  |   ALGs+CD        0x0002           32768

switch>
```

show lacp internal

The **show lacp internal** command displays the local LACP state for all specified channels. Local state data includes the state machines and LACP protocol information.

Command Mode
EXEC

Command Syntax
show lacp [***PORT_LIST***] **internal** [***PORT_LEVEL***] [***INFO_LEVEL***]

Parameters

- *PORT_LIST* interface for which port information is displayed. Options include:
  - <no parameter> all configured port channels
  - *c_range* ports in specified channel list (number, number range, or list of numbers and ranges).
  - **interface** ports on all interfaces.
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* loopback interface specified by *l_num*.
  - **interface management** *m_num* management interface specified by *m_num*.
  - **interface port-channel** *p_num* port channel interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.
  - **interface peerethernet***pe_num* peer Ethernet interface specified by p*e_num*.
  - **interface peerport-channel***pc_num* peer port channel interface specified by *pc_num*.
- *PORT_LEVEL* ports displayed, in terms of aggregation status. Options include:
  - <no parameter> command lists data for ports bundled by LACP into an aggregate.
  - **all-ports** command lists data for all ports, including LACP candidates that are not bundled.
- *INFO_LEVEL* amount of information that is displayed. Options include:
  - <no parameter> displays same information as **brief** option.
  - **brief** displays LACP configuration data, including sys-id, actor, priorities, and keys.
  - **detailed** includes **brief** option information plus state machine data.

PORT_LEVEL and *INFO_LEVEL* parameters can be placed in any order.

Example

- This command displays internal data for all configured port channels.

```
switch>show lacp internal
LACP System-identifier: 8000,00-1c-73-04-36-d7
State: A = Active, P = Passive; S=ShortTimeout, L=LongTimeout;
       G = Aggregable, I = Individual; s+=InSync, s-=OutOfSync;
       C = Collecting, X = state machine expired,
       D = Distributing, d = default neighbor state
             |Partner                                  Actor
Port Status  | Sys-id                Port#  State     OperKey  PortPriority
-------------------------------------------------------------------------------
Port Channel Port-Channel1:
Et43 Bundled | 8000,00-1c-73-09-a0-f3    43  ALGs+CD   0x0001         32768
Et44 Bundled | 8000,00-1c-73-09-a0-f3    44  ALGs+CD   0x0001         32768
Et45 Bundled | 8000,00-1c-73-09-a0-f3    45  ALGs+CD   0x0001         32768
Et46 Bundled | 8000,00-1c-73-09-a0-f3    46  ALGs+CD   0x0001         32768
```

show lacp neighbor

The **show lacp neighbor** command displays the LACP protocol state of the remote neighbor for all specified port channels.

Command Mode

EXEC

Command Syntax

show lacp [***PORT_LIST***] **neighbor** [***PORT_LEVEL***] [***INFO_LEVEL***]

PORT_LEVEL and ***INFO_LEVEL*** parameters can be placed in any order.

Parameters

- ***PORT_LIST*** interface for which port information is displayed. Options include:
  - <no parameter> displays information for all configured port channels
  - *c_range* ports in specified channel list (number, number range, or list of numbers and ranges).
  - **interface** ports on all interfaces.
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* loopback interface specified by *l_num*.
  - **interface management** *m_num* management interface specified by *m_num*.
  - **interface port-channel** *p_num* port channel interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.
  - **interface peerethernet***pe_num* peer Ethernet interface specified by p*e_num*.
  - **interface peerport-channel***pc_num* peer port channel interface specified by *pc_num*.
- ***PORT_LEVEL*** ports displayed, in terms of aggregation status. Options include:
  - <no parameter> command lists data for ports bundled by LACP into an aggregate.
  - **all-ports** command lists data for all ports, including LACP candidates that are not bundled.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - <no parameter> displays same information as **brief** option.
  - **brief** displays LACP configuration data, including sys-id, actor, priorities, and keys.
  - **detailed** includes **brief** option information plus state machine data.

Example

- This command displays the LACP protocol state of the remote neighbor for all port channels.

```
switch>show lacp neighbor
State: A = Active, P = Passive; S=ShortTimeout, L=LongTimeout;
       G = Aggregable, I = Individual; s+=InSync, s-=OutOfSync;
       C = Collecting, X = state machine expired,
       D = Distributing, d = default neighbor state
               |                            Partner
Port   Status  | Sys-id                   Port#   State     OperKey  PortPri
-------------------------------------------------------------------------------
Port Channel Port-Channel1:
Et1    Bundled | 8000,00-1c-73-00-13-19      1   ALGs+CD    0x0001    32768
Et2    Bundled | 8000,00-1c-73-00-13-19      2   ALGs+CD    0x0001    32768
Port Channel Port-Channel2:
Et23   Bundled | 8000,00-1c-73-04-36-d7     47   ALGs+CD    0x0002    32768
Et24   Bundled | 8000,00-1c-73-04-36-d7     48   ALGs+CD    0x0002    32768
Port Channel Port-Channel4*:
Et3    Bundled | 8000,00-1c-73-0b-a8-0e     45   ALGs+CD    0x0001    32768
Et4    Bundled | 8000,00-1c-73-0b-a8-0e     46   ALGs+CD    0x0001    32768
Port Channel Port-Channel5*:
Et19   Bundled | 8000,00-1c-73-0c-30-09     49   ALGs+CD    0x0005    32768
Et20   Bundled | 8000,00-1c-73-0c-30-09     50   ALGs+CD    0x0005    32768
Port Channel Port-Channel6*:
Et6    Bundled | 8000,00-1c-73-01-07-b9     49   ALGs+CD    0x0001    32768
Port Channel Port-Channel7*:
Et5    Bundled | 8000,00-1c-73-0f-6b-22     51   ALGs+CD    0x0001    32768
Port Channel Port-Channel8*:
Et10   Bundled | 8000,00-1c-73-10-40-fa     51   ALGs+CD    0x0001    32768

* - Only local interfaces for MLAGs are displayed. Connect to the peer to
    see the state for peer interfaces.
switch>
```

show lacp sys-id

The **show lacp sys-id** command displays the System Identifier the switch uses when negotiating remote LACP implementations.

Command Mode

EXEC

Command Syntax

show lacp sys-id [***INFO_LEVEL***]

Parameters

- *INFO_LEVEL* amount of information that is displayed. Options include:
  - <no parameter> displays system identifier
  - **brief** displays system identifier.
  - **detailed** displays system identifier and system priority, including the MAC address.

Examples

- This command displays the system identifier.

```
switch>show lacp sys-id brief
8000,00-1c-73-04-36-d7
```

- This command displays the system identifier and system priority.

```
switch>show lacp sys-id detailed
System Identifier used by LACP:
System priority: 32768  Switch MAC Address: 00:1c:73:04:36:d7
  802.11.43 representation: 8000,00-1c-73-04-36-d7
```

show port-channel

The **show port-channel** command displays information about members the specified port channels.

Command Mode

EXEC

Command Syntax

```
show port-channel [MEMBERS] [PORT_LIST] [INFO_LEVEL]
```

Parameters

- *MEMBERS* list of port channels for information is displayed. Options include:
  - <no parameter> all configured port channels.
  - *p_range* ports in specified channel list (number, number range, or list of numbers and ranges).
- *PORT_LEVEL* ports displayed, in terms of aggregation status. Options include:
  - <no parameter> Displays information on ports that are active members of the LAG.
  - **active-ports** Displays information on ports that are active members of the LAG.
  - **all-ports** Displays information on all ports (active or inactive) configured for LAG.
- *INFO_LEVEL* amount of information that is displayed. Options include:
  - <no parameter> Displays information at the brief level.
  - **brief** Displays information at the brief level.
  - **detail** Displays information at the detail level.

Display Values

- **Port Channel** Type and name of the port channel.
- **Time became active** Time when the port channel came up.
- **Protocol** Protocol operating on the port.
- **Mode** Status of the Ethernet interface on the port. The status value is Active or Inactive.
- **No active ports** Number of active ports on the port channel.
- **Configured but inactive ports** Ports configured but that are not actively up.
- **Reason unconfigured** Reason why the port is not part of the LAG.

Guidelines

The **show etherchannel** and **show port-channel** commands are identical.

You can configure a port channel to contain many ports, but only a subset may be active at a time. All active ports in a port channel must be compatible. Compatibility includes many factors and is platform specific. For example, compatibility may require identical operating parameters such as speed and maximum transmission unit (MTU). Compatibility may only be possible between specific ports because of the internal organization of the switch.

Examples

- This command displays output from the **show port-channel** command:

```
switch>show port-channel 3
Port Channel Port-Channel3:
  Active Ports:
       Port                Time became active       Protocol    Mode
     ------------------------------------------------------------------------
       Ethernet3           15:33:41                 LACP        Active
       PeerEthernet3       15:33:41                 LACP        Active
```

- This command displays output from the **show port-channel active-ports** command:

```
switch>show port-channel active-ports
Port Channel Port-Channel3:
  No Active Ports
Port Channel Port-Channel11:
  No Active Ports
```

- This command displays output from the **show port-channel all-ports** command:

```
switch>show port-channel all-ports
Port Channel Port-Channel3:
  No Active Ports
  Configured, but inactive ports:
      Port            Time became inactive    Reason unconfigured

-------------------------------------------------------------------------------
      Ethernet3       Always                  not compatible with aggregate

Port Channel Port-Channel11:
  No Active Ports
  Configured, but inactive ports:
      Port            Time became inactive    Reason unconfigured
-------------------------------------------------------------------------------
      Ethernet25      Always                  not compatible with aggregate
      Ethernet26      Always                  not compatible with aggregate
```

show port-channel limits

The **show port-channel limits** command displays groups of ports that are compatible and may be joined into port channels. Each group of compatible ports is called a LAG group. For each LAG group, the command also displays ***Max interfaces*** and ***Max ports per interface***.

- ***Max interfaces*** defines the maximum number of active port channels that may be formed out of these ports.
- ***Max ports per interface*** defines the maximum number of active ports allowed in a port channel from the compatibility group.

All active ports in a port channel must be compatible. Compatibility comprises many factors and is specific to a given platform. For example, compatibility may require identical operating parameters such as speed and/or maximum transmission unit (MTU). Compatibility may only be possible between specific ports because of internal organization of the switch.

Command Mode

EXEC

Command Syntax

```
show port-channel limits
```

Example

- This command displays **show port-channel list** output:

```
switch>show port-channel limits
LAG Group: focalpoint
------------------------------------------------------------------------
  Max port-channels per group: 24, Max ports per port-channel: 16
  24 compatible ports: Ethernet1  Ethernet2  Ethernet3  Ethernet4
                       Ethernet5  Ethernet6  Ethernet7  Ethernet8
                       Ethernet9  Ethernet10 Ethernet11 Ethernet12
                       Ethernet13 Ethernet14 Ethernet15 Ethernet16
                       Ethernet17 Ethernet18 Ethernet19 Ethernet20
                       Ethernet21 Ethernet22 Ethernet23 Ethernet24
------------------------------------------------------------------------

switch>
```

show port-channel load-balance fields

The **show port-channel load-balance fields** command displays the fields that the hashing algorithm uses to distribute traffic across the interfaces that comprise the port channels.

Command Mode

EXEC

Command Syntax

show port-channel load-balance *HARDWARE* fields

Parameters

- *HARDWARE* ASIC switching device. Selection options depend on the switch model and include:
  - **fm4000**
  - **petraA**
  - **trident**

Examples

- This command displays the hashing fields used for balancing port channel load.

```
switch>show port-channel load-balance fm4000 fields
Source MAC address hashing for non-IP packets is ON
Destination MAC address hashing for non-IP packets is ON
Ethernet type hashing for non-IP packets is ON
Source MAC address hashing for IP packets is ON
Destination MAC address hashing for IP packets is ON
Ethernet type hashing for IP packets is ON
IP source address hashing is ON
IP destination address hashing is ON
IP protocol field hashing is ON
TCP/UDP source port hashing is ON
TCP/UDP destination port hashing is ON
switch>
```

show port-channel summary

The **show port-channel summary** command displays the port-channels on the switch and lists their component interfaces, LACP status, and set flags**.**

Command Mode

EXEC

Command Syntax

show port-channel summary

Examples

- This command displays **show port-channel summary** output:

```
switch>show port-channel summary

    Flags
--------------------------------------------------------------------------------
a - LACP Active            p - LACP Passive
U - In Use                 D - Down
+ - In-Sync                - - Out-of-Sync          i - incompatible with agg
P - bundled in Po          s - suspended            G - Aggregable
I - Individual             S - ShortTimeout         w - wait for agg

Number of channels in use: 2
Number of aggregators:2

   Port-Channel       Protocol    Ports
------------------------------------------------------
   Po1(U)             LACP(a)      Et47(PG+) Et48(PG+)
   Po2(U)             LACP(a)      Et39(PG+) Et40(PG+)
```

show port-channel traffic

The **show port-channel traffic** command displays the traffic distribution between the member ports of the specified port channels. The command displays distribution for unicast, multicast, and broadcast streams.

Command Mode

EXEC

Command Syntax

show port-channel [***MEMBERS***] **traffic**

Parameters

- *MEMBERS* list of port channels for which information is displayed. Options include:
  - <no parameter> all configured port channels.
  - *c_range* ports in specified channel list (number, number range, or list of numbers and ranges).

Examples

- This command displays traffic distribution for all configured port channels.

```
switch>show port-channel traffic
ChanId      Port Rx-Ucst Tx-Ucst Rx-Mcst Tx-Mcst Rx-Bcst Tx-Bcst
------ --------- ------- ------- ------- ------- ------- -------
     8      Et10 100.00% 100.00% 100.00% 100.00%   0.00% 100.00%
------ --------- ------- ------- ------- ------- ------- -------
     1       Et1  13.97%  42.37%  47.71%  30.94%   0.43%  99.84%
     1       Et2  86.03%  57.63%  52.29%  69.06%  99.57%   0.16%
------ --------- ------- ------- ------- ------- ------- -------
     2      Et23  48.27%  50.71%  26.79%  73.22%   0.00% 100.00%
     2      Et24  51.73%  49.29%  73.21%  26.78%   0.00%   0.00%
------ --------- ------- ------- ------- ------- ------- -------
     4       Et3  55.97%  63.29%  51.32%  73.49%   0.00%   0.00%
     4       Et4  44.03%  36.71%  48.68%  26.51%   0.00%   0.00%
------ --------- ------- ------- ------- ------- ------- -------
     5      Et19  39.64%  37.71%  50.00%  90.71%   0.00%   0.00%
     5      Et20  60.36%  62.29%  50.00%   9.29%   0.00% 100.00%
------ --------- ------- ------- ------- ------- ------- -------
     6       Et6 100.00% 100.00% 100.00% 100.00%   0.00% 100.00%
------ --------- ------- ------- ------- ------- ------- -------
     7       Et5 100.00%   0.00% 100.00% 100.00%   0.00%   0.00%
switch>
```

Chapter 10

Data Transfer

Arista switches support the transfer of packets (network layer) and frames (data link layer). This chapter describes concepts and processes that are referenced by routing and switching protocols that Arista switches support.

Sections in this chapter include:

- Section 10.1: Introduction
- Section 10.2: Data Transfer Methods
- Section 10.3: Ports
- Section 10.4: Routing Concepts
- Section 10.5: Data Transfer Configuration Processes
- Section 10.6: Data Transfer Command Descriptions

10.1 Introduction

Arista switch transfer data through switching, routing, and layer 3 switching. This chapter provides an introduction to these transfer methods.

Data structures and supporting processes that support data transfer methods and referenced in specific protocol chapters are also described, including:

- routed ports
- port mirroring
- storm control
- loopback interfaces
- route redistribution
- Equal Cost Multipath routing (ECMP)
- null0 interfaces
- MTUs
- routing tables
- VPN Routing and Forwarding (VRFs).

10.2 Data Transfer Methods

This section describes these data transfer methods:

- Section 10.2.1: Switching and Bridging
- Section 10.2.2: Routing
- Section 10.2.3: Layer 3 Switching

10.2.1 Switching and Bridging

Switching and bridging operations transmit data link layer frames between devices within a single subnet. Each port is assigned a 48 bit Media Access Control (MAC) address. Frames arriving at a hub are bridged, or sent to all other ports on the subnet. Switches can associate ports with their MAC addresses, obviating the need to flood the subnet when sending a frame.

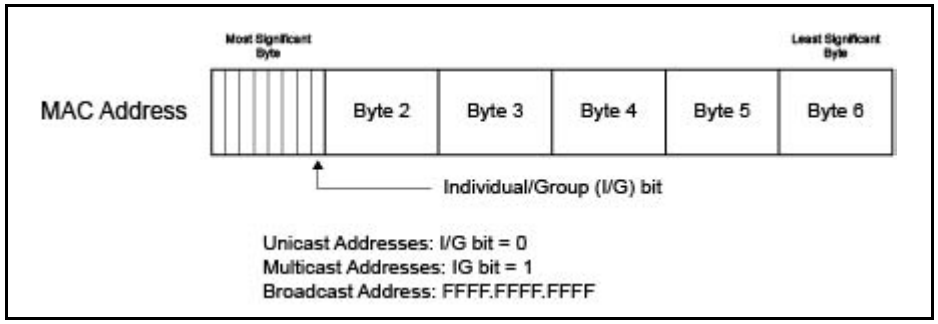
Subnets in the switch are defined by VLANs. A virtual local area network (VLAN) is a group of devices that are configured to communicate as if they are attached to the same network regardless of their physical location. Chapter 11 describes VLANS.

Four MAC address types identify the scope of LAN interfaces that an address represents:

- unicast: represents a single interface.
- broadcast: represents all interfaces.
- multicast: represents a subset of all interfaces.
- reserved: assigned to nodes that have no configured MAC address.

The Individual/Group (I/G) bit distinguishes unicast MAC addresses from multicast addresses. As shown in Figure 10-1, the I/G bit is the least significant bit of the most significant byte in a MAC address.

Figure 10-1 MAC Address Format



- Unicast address: the I/G bit is 0: 1234.1111.1111 is a unicast MAC address (the most significant byte is an even number).

  Reserved address: all bits set to 0 (0000.0000.0000).

- Multicast address: the I/G bit is 1: 1134.1111.1111 is a multicast MAC address (the most significant byte is an odd number).

  Broadcast address: all bits set to 1 (FFFF.FFFF.FFFF).

  Examples

  - The following are unicast MAC addresses:

```
0200.0000.0000
1400.0000.0000
```

- The following are multicast MAC addresses:

```
0300.0000.0000
2500.0000.0000
```

The following sections describe MAC address functions and data structures.

- Section 8.5.2 describes the process of assigning a MAC address to an interface.
- Section 10.3.4 describes the MAC Address table.

10.2.2 Routing

Routing transmits network layer packets over connected independent subnets. Each subnet is assigned an IP address range and each device on the subnet is assigned an IP address from that range. Connected subnets have IP address ranges that do not overlap. A router connects multiple subnets. Routers forward inbound packets to the subnet whose address range includes the packets' destination address.

IPv4 and IPv6 are internet layer protocols that facilitate packet-switched networking, including transmissions across multiple networks.

These chapters describe available IP features:

- **IPv4**: Chapter 17
- **IPv6**: Chapter 18

10.2.2.1 Static Routing

Static routes are entered through the CLI and are typically used when dynamic protocols are unable to establish routes to a specified destination prefix. Static routes are also useful when dynamic routing protocols are not available or appropriate.

Creating a static route associates a destination IP address with a local interface. The routing table refers to these routes as ***connected*** routes that are available for redistribution into routing domains defined by dynamic routing protocols.

These sections static route configuration commands:

- Section 17.3.2.2: Creating Static and Default IPv4 Routes
- Section 18.3.1.2: Configuring Default and Static IPv6 Routes

10.2.2.2 Dynamic Routing

Dynamic routes are established by dynamic routing protocols. These protocols also maintain the routing table and modify routes to adjust for topology or traffic changes. Routing protocols assist the switch in communicating with other devices to exchange network information, maintaining routing tables, and establishing data paths.

The switch supports these dynamic routing protocols:

- **Open Shortest Path First – Version 2**: Chapter 19
- **Open Shortest Path First – Version 3**: Chapter 20
- **Border Gateway Protocol**: Chapter 21
- **Routing Information Protocol**: Chapter 22

10.2.3 Layer 3 Switching

Layer 3 switches establish data paths through routing processes (Layer 3) and transfer data as a switch (Layer 2) through speed-optimized hardware. Layer 3 switches use a control plane (routing) and data plane (switching) to manage these processes.

10.2.3.1 Control plane

The control plane builds and maintains the IP routing table, which identifies IP packet routes in terms of destination addresses. The routing table defines a route by its next hop address and the egress interface that accesses the next hop.

The control plane derives routing information from three sources:

- Status of physical and virtual interfaces on the switch.
- Static routes entered through the CLI.
- Routes established through dynamic routing protocols.

Applying an ACL to the Control Plane (page 455) describes control plane access control lists.

10.2.3.2 Data plane

The data plane routes IP packets based on information derived by the control plane. Each packet's path includes Layer 2 addresses that reaches its next hop destination. The data plane also performs other operations required by IP routing, such as recalculating IP header checksums and decrementing the time-to-live (TTL) field.

Arista data planes support these packet forwarding modes:

- Store and forward: the switch accumulates entire packets before forwarding them.
- Cut through: the switch begins forwarding frames before their reception is complete.

Cut through mode reduces switch latency at the risk of decreased reliability. Packet transmissions can begin immediately after the destination address is processed. Corrupted frames may be forwarded because packet transmissions begin before CRC bytes are received.

Packet forwarding mode availability varies by switch platform:

- FM4000: cut-through mode only
- Petra: store and forward mode only
- Trident: both modes are available.

Section 10.5.4 describes the configuration of the forwarding plane mode.

10.3 Ports

This section describes these port properties:

- Section 10.3.1: Port Mirroring
- Section 10.3.2: Storm Control
- Section 10.3.3: Switched and Routed Ports
- Section 10.3.4: MAC Address Table
- Section 10.3.5: Loopback Ports
- Section 10.3.6: Port Security
- Section 10.3.7: Null0 Interface
- Section 10.3.8: Maximum Transmission Units (MTU)

10.3.1 Port Mirroring

Port mirroring, also known as port monitoring, is the duplication of traffic from a set of source ports onto a destination port. A mirror session associates a set of source ports to a destination port.

Mirror sources can be Ethernet or port channel interfaces. Mirror destination ports are Ethernet interfaces. Mirror ports cannot be routed ports or Ethernet ports that are members of a port channel. Mirroring is inactivated on Ethernet mirror ports that are subsequently added to a port channel.

Layer 2 control protocols run normally on source ports; PDU traffic is mirrored identically to data traffic. Layer 2 control protocols do not run on destination ports.

An interface cannot be used in a more than one mirror session and cannot be simultaneously a source and destination. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from one direction.

- Ingress Mirroring: Packets received by a source port are duplicated, including all valid data frames and L2 control PDUs. The port mirrors data before forwarding logic is applied. Packets subsequently dropped because of forwarding decisions are mirrored.
- Egress Mirroring: Packets transmitted by a source port are duplicated, with these exceptions:
  - Flooded/Multicast Packets: Packets sent to multiple mirror ports generate one copy. The exception is when the mirror source and mirror destination are on different chips in a multi-chip device; in this case, an extra copy is generated.
  - Dropped Packets: Packets dropped by forwarding decisions (such output STP state checks) on egress sources are not duplicated. Packets dropped because of congestion may be duplicated.

  VLAN tagging on duplicate packets from an egress source is identical to the tagging of the inbound source packet.

When a packet's path through the switch includes multiple mirror source ports in different mirror sessions, the traffic is duplicated once and sent to the destination of the highest numbered session.

Section 10.5.2.2: Mirror Port Configuration describes port mirroring configuration procedures.

10.3.2 Storm Control

A traffic storm is a flood of packets entering a network, resulting in excessive traffic and degraded performance. Storm control prevents network disruptions by limiting traffic beyond specified thresholds on individual physical LAN interfaces.

Storm control monitors inbound traffic levels over a one-second intervals and compares the traffic level with a specified benchmark. The storm control level is a percentage of the total available bandwidth of the port and is configurable for multicast and broadcast packets on each interface.

Storm control provides three mode options

- Storm control all: When inbound traffic exceeds the specified threshold within a one-second control interval, all traffic is dropped until the end of the interval.
- Storm control broadcast: When inbound broadcast traffic exceeds the specified threshold within a one-second control interval, broadcast traffic is dropped until the end of the interval.
- Storm control multicast: When inbound multicast traffic exceeds the specified threshold within a one-second control interval, multicast traffic is dropped until the end of the interval.

Broadcast and multicast storm control are independent features and can be enabled simultaneously. The storm control all threshold overrides broadcast and multicast thresholds.

Section 10.5.2.3: Storm Control Configuration describes storm control configuration procedures.

10.3.3 Switched and Routed Ports

A switched port is an Ethernet or port channel interface that is configured as a layer 2 interface. Switched ports bridge frames and are assigned to at least one VLAN. Switched ports are not associated with any IP addresses. By default, Ethernet and port channel interfaces are in switched port mode.

A routed port is an Ethernet or port channel interface that is configured as a layer 3 interface. Routed ports do not bridge frames and are not members of any VLANs. Routed ports can have IP addresses assigned to them and packets are routed directly to and from the port.

Configuring an interface as a routed port is similar to creating a VLAN with spanning-tree disabled, making the port the only member of that VLAN and configuring the IP address on the switch virtual interface (SVI) associated with the VLAN.

All IP-level interface configuration commands, except **autostate** and **ip virtual-router,** can be used to configure a routed interface. If the interface is reverted to switched port mode, ***running-config*** maintains IP level interface configuration statements. These changes become active again if the interface is configured back to routed port mode.

A LAG that is created with the **channel-group** command inherits the mode of the member port. A LAG created from a routed port becomes a routed LAG. IP-level configuration is not propagated to the LAG from its component members.

Section 10.5.2.1: Routed Port Configuration describes the port mode configuration procedure.

10.3.4 MAC Address Table

The switch maintains an MAC address table for switching frames efficiently between ports. The MAC address table contains static and dynamic MAC addresses.

- Dynamic MAC addresses are entered into the table when the switch receives a frame whose source address is not listed in the MAC address table.

  The switch builds the table dynamically by referencing the source address of frames it receives.
- Static MAC addresses are entered into the table through a CLI command.

When the switch receives a frame, it associates the MAC address of the transmitting interface with the recipient VLAN. When a VLAN receives a frame for a MAC destination address not listed in the address table, the switch bridges the frame to all of the VLAN's ports except the recipient port. When the destination interface replies, the switch adds its MAC address address to the MAC address table. The switch forwards subsequent frames with the destination adddress to the specified port.

A multicast address can be associated with multiple ports.

10.3.5 Loopback Ports

A loopback interface is a virtual network interface implemented in software and does not connect to any hardware. Traffic sent to the loopback interface is immediately received on the sending interface.

Internet protocols reserve specific addresses for loopback network segments:

- IPv4 designates 127/8 as loopback subnet, which includes 127.0.0.0 through 127.255.255.255.
- IPv6 designates ::1/128 as the loopback address, which includes 0:0:0:0:0:0:0:1 (also written as ::1).

Arista switches support the configuration of 1001 loopback interfaces, numbered from 0 to 1000.

Section 10.5.2.5: Loopback Interface Configuration Mode describes loopback interface configuration procedures.

10.3.6 Port Security

Port security restricts input to a switched port by limiting the number and identity of MAC addresses that can access the port. By limiting a port's access to a single specific MAC address, the host assigned to that address is guaranteed full bandwidth of the port. A secure port drops frames received from MAC addresses whose inclusion in the MAC address table exceeds the port-security maximum.

Section 10.5.2.4: Port Security Configuration describes the port security configuration procedures.

10.3.7 Null0 Interface

The null0 interface is a virtual interface that drops all inbound packets. A null0 route is a network route whose destination is null0 interface and inbound packets are not forwarded to any valid address.

Many interface configuration commands provide ***null0*** as an interface option.

10.3.8 Maximum Transmission Units (MTU)

The maximum transmission unit (MTU) of a communications protocol denotes the size (bytes) of the largest packet that can be sent over the network. Different protocols support a variety of MTU sizes. Most IP over Ethernet implementations use the Ethernet V2 frame format, which specifies an MTU of 1500 bytes. Jumbo frames are Ethernet frames containing more than 1500.

MTU is independently configurable on all routable interfaces. The switch supports MTU sizes ranging from 68 to 9212 bytes. The default MTU is 1500 bytes.

Section 10.5.2.6: MTU Configuration describes MTU configuration procedures.

10.4 Routing Concepts

When routing is enabled, the switch discovers the best route to a packet's destination address by exchanging routing information with other devices. IP routing is disabled by default.

The following sections describes routing features that the switch supports

- Section 10.4.1: Route Redistribution
- Section 10.4.2: Equal Cost Multipath Routing (ECMP) and Load Sharing
- Section 10.4.3: Routing Tables / VPN Routing and Forwarding (VRF)

10.4.1 Route Redistribution

Route redistribution is the advertisement, into a dynamic routing protocol's routing domain, of connected (static) routes or routes established by other routing protocols. By default, the switch advertises only routes in a routing domain that are established by the protocol that defined the domain.

Route redistribution commands specify the scope of the redistribution action. By default, all routes from a specified protocol (or all static routes) are advertised into the routing domain. Commands can also filter routes by applying a route map, which defines the subset of routes to be advertised.

10.4.2 Equal Cost Multipath Routing (ECMP) and Load Sharing

Multiple routes that have identical destinations and administrative distances comprise an Equal Cost Multi-Path (ECMP) route. The switch attempts to spread traffic to all ECMP route paths equally.

If two or more equal-cost paths from a source are available, unicast traffic is load split across those paths. By default, multicast traffic is not load split. Multicast traffic generally flows from the reverse path forwarding (RPF) neighbor and, according to Protocol Independent Multicast (PIM) specifications, the neighbor with the highest IP address has precedence when multiple neighbors have the same metric.

Section 17.3.2.3: Configuring ECMP (IPv4) describes ECMP configuration procedures for IPv4.

10.4.3 Routing Tables / VPN Routing and Forwarding (VRF)

An IP routing table is a data table that lists the routes to network destinations metrics (distances) associated with those routes. A routing table is also known as a routing information base (RIB).

VPN Routing and Forwarding (VRF) allows traffic separation by maintaining multiple routing tables. Arista switches support two VRF instances: a global or default VRF called "main" and a user defined VRF. The user defined VRF can be used as a management VRF to segregate management traffic from customer traffic. Routing cannot be performed within a user defined VRF.

To allow overlap in the sets of IP addresses used by different VRF instances, a route distinguisher (RD) is prepended to each address. RDs are defined in RFC 4364.

Section 10.5.5: Configuring VRFs describes VRF configuration procedures.

10.5 Data Transfer Configuration Processes

These sections describe basic data transfer configuration tasks.

- Section 10.5.1: MAC Address Table Configuration
- Section 10.5.2: Port Configuration
- Section 10.5.4: Data Plane Forwarding Mode
- Section 10.5.5: Configuring VRFs

10.5.1 MAC Address Table Configuration

10.5.1.1 Static MAC Address Table Entries

The MAC address table accepts static MAC addresses, including multicast entries. Each table entry references a MAC address, a VLAN, and a list of layer 2 (Ethernet or port channel) ports. The table supports three entry types: unicast drop, unicast, and multicast.

- A drop entry does not include a port.
- A unicast entry includes one port.
- A multicast entry includes at least one port.

  Packets with a MAC address (source or destination) and VLAN specified by a drop entry are dropped. Drop entries are valid for only unicast MAC addresses.

The **mac address-table static** command adds a static entry to the MAC address table.

Example

- This command adds the static entry for unicast MAC address 0012.3694.03ec to the MAC address table.

```
switch(config)#mac address-table static 0012.3694.03ec vlan 3 interface
Ethernet 7
switch(config)#show mac address-table static
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
   3    0012.3694.03ec    STATIC      Et7
Total Mac Addresses for this criterion: 1

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0

switch(config)#
```

- This command adds the static entry for the multicast MAC address 0112.3057.8423 to the MAC address table.

```
switch(config)#mac address-table static 0112.3057.8423 vlan 4 interface
port-channel 10 port-channel 12
switch(config)#show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
Total Mac Addresses for this criterion: 0

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
   4    0112.3057.8423    STATIC      Po10 Po12
Total Mac Addresses for this criterion: 1
switch(config)#
```

10.5.1.2 Dynamic MAC Address Table Entries

Learning Mode

The switch maintains an MAC address table for switching frames efficiently between VLAN ports. When the switch receives a frame, it associates the MAC address of the transmitting interface with the recipient VLAN and port. When MAC address learning is enabled for the recipient port, the entry is added to the MAC address table. When MAC address learning is not enabled, the entry is not added to the table.

The **switchport mac address learning** command enables MAC address learning for the configuration mode interface. MAC address learning is enabled by default on all Ethernet and port channel interfaces.

Example

- These commands disables MAC address learning for Ethernet interface 8, then displays the active configuration for the interface.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#no switchport mac address learning
switch(config-if-Et8)#show active
interface Ethernet8
   no switchport mac address learning
switch(config-if-Et8)#
```

Aging Time

Aging time defines the period an entry is in the table, as measured from the most recent reception of a frame on the entry's VLAN from the specified MAC address. The switch removes entries when their presence in the MAC address table exceeds the aging time.

Aging time ranges from 10 to 1,000,000 seconds with a default of 300 seconds (five minutes).

The **mac address-table aging-time** command configures the aging time for MAC address table dynamic entries. Aging time defines the period an entry is in the table, as measured from the most recent reception of a frame on the entry's VLAN from the specified MAC address. The switch removes entries when their presence in the MAC address table exceeds the aging time.

Clearing Dynamic Addresses

The **clear mac address-table dynamic** command removes specified dynamic entries from the MAC address table. Entries are identified by their VLAN and layer 2 (Ethernet or port channel) interface.

Example

- This command clears all dynamic mac address table entries for port channel 5 on VLAN 34.

```
Switch(config)#clear mac address-table dynamic vlan 34 interface
port-channel 5
Switch(config)
```

10.5.1.3 Displaying the MAC Address Table

The **show mac address-table** command displays the specified MAC address table entries.

Example

- This command displays the MAC address table.

```
Switch#show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
 101    001c.8224.36d7    DYNAMIC     Po2        1       9 days, 15:57:28 ago
 102    001c.8220.1319    STATIC      Po1
 102    001c.8229.a0f3    DYNAMIC     Po1        1       0:05:05 ago
 661    001c.8220.1319    STATIC      Po1
 661    001c.822f.6b22    DYNAMIC     Po7        1       0:20:10 ago
3000    001c.8220.1319    STATIC      Po1
3000    0050.56a8.0016    DYNAMIC     Po1        1       0:07:38 ago
3909    001c.8220.1319    STATIC      Po1
3909    001c.822f.6a80    DYNAMIC     Po1        1       0:07:08 ago
3910    001c.730f.6a80    DYNAMIC     Et9        1       4 days, 15:13:07 ago
3911    001c.8220.1319    STATIC      Po1
3911    001c.8220.40fa    DYNAMIC     Po8        1       1:19:58 ago
3912    001c.822b.033e    DYNAMIC     Et11       1       9 days, 15:57:23 ago
3913    001c.8220.1319    STATIC      Po1
3913    001c.822b.033e    DYNAMIC     Po1        1       0:04:35 ago
3984    001c.8220.178f    DYNAMIC     Et8        1       4 days, 15:07:29 ago
3992    001c.8220.1319    STATIC      Po1
3992    001c.8221.07b9    DYNAMIC     Po6        1       4 days, 15:13:15 ago
Total Mac Addresses for this criterion: 25

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0
Switch#
```

10.5.2 Port Configuration

This section describes the following port and interface configuration tasks:

- Section 10.5.2.1: Routed Port Configuration
- Section 10.5.2.2: Mirror Port Configuration
- Section 10.5.2.3: Storm Control Configuration
- Section 10.5.2.4: Port Security Configuration
- Section 10.5.2.5: Loopback Interface Configuration Mode
- Section 10.5.2.6: MTU Configuration

10.5.2.1 Routed Port Configuration

The switching-routing configuration of Ethernet and port channel interfaces is specified by the **switchport** command:

- The **switchport** command places the configuration mode interface in ***switched port*** (Layer 2) mode. Switched ports are configurable as members of one or more VLANs through other switchport commands. Switched ports ignore all IP level configuration commands, including IP address assignments. By default, Ethernet and port channel interfaces are switched ports
- The **no switchport** command places the configuration mode interface in ***routed port*** mode. Routed ports behave as Layer 3 interfaces. They do not bridge packets and are not VLAN members. An IP address can be assigned to a routed port for the direct routing of packets to and from the interface.

  These commands only toggle the interface between switched and routed modes. They have no effect on other configuration states.

The **default switchport** command reverts the the configuration mode interface to switched port mode by removing the corresponding **no switchport** command from ***running-config***.

When an interface is configured as a routed port, the switch transparently allocates an internal VLAN whose only member is the routed interface. Internal VLANs are created in the range from 1006 to 4094. VLANs that are allocated internally for a routed interface cannot be directly created or configured. Section 11.3.5: Allocating Internal VLANs describes VLAN allocation configuration procedures.

Examples

- These commands put Ethernet interface 5 in routed port mode.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#no switchport
```

- These commands returns Ethernet interface 5 to switched port mode.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#switchport
```

10.5.2.2 Mirror Port Configuration

Port mirroring is the duplication of traffic from a set of source ports onto a destination port. A mirror session associates a set of source ports to a destination port.

- The **monitor session source** command configures an Ethernet or port channel interface as the source port of a specified port mirroring session.
- The **monitor session destination** command configures an Ethernet interface as the destination port of a specified port mirroring session.

An interface cannot be used in a more than one mirror session and cannot be simultaneously a source and destination. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from only one direction.

Example

- These commands configures Ethernet interface 8 as the source port and Ethernet interface 7 as the destination port for the ***redirect_1*** mirroring session.

```
switch(config)#monitor session redirect_1 source ethernet 7
switch(config)#monitor session redirect_1 destination ethernet 8
```

- This command displays the configuration of the ***redirect_1*** mirroring session.

```
switch(config)#show monitor session

Session redirect_1
------------------------

Source Ports

  Both:        Et7

Destination Port: Et8

switch(config)#
```

10.5.2.3 Storm Control Configuration

The **storm-control** command configures and enables broadcast or multicast storm control on the configuration mode interface. The command provides three mode options:

- **storm-control all** all inbound packet control.
- **storm-control broadcast** broadcast inbound packet control.
- **storm-control multicast** multicast inbound packet control.

An interface configuration can contain three storm-control statements, one with each mode setting. The **storm-control all** threshold overrides broadcast and multicast thresholds.

When storm control is enabled, the switch monitors inbound traffic levels over one second intervals and compares the traffic level with a specified threshold. The threshold is a percentage of the total available port bandwidth and is configurable on each interface for each transmission mode.

Example

- This command enables multicast storm control on Ethernet interface 3 and sets a threshold of 65%. During each one second interval, the interface drops multicast traffic it receives in excess of 65% of the port capacity.

```
Switch(config)#interface ethernet 3
Switch(config-if-Et3)#storm-control multicast level 65
Switch(config-if-Et3)#
```

The **show storm-control** command displays the storm-control level and interface inbound packet capacity for the specified interface.

Example

- This command displays the storm control configuration for Ethernet ports 1 through 5.

```
Switch(config-if-Et3)#show storm-control ethernet 1-5
Port BcastEnabled BcastLevel BcastRate(Mbps) McastEnabled McastLevel McastRate(Mbps)
Et1            No        100               -          No        100               -
Et2            No        100               -          No        100               -
Et3            No        100               -         Yes         29            2976
Et4           Yes         29            2976         Yes         29            2976
Et5            No        100               -          No        100               -
```

10.5.2.4 Port Security Configuration

Port security restricts input to a switched port by limiting the number and identity of MAC addresses that can access the port.

MAC address port security is enabled by the **switchport port-security** command. Ports with port security enabled restrict traffic to a limited number of hosts, as determined by their MAC addresses. The maximum number of MAC addresses that can be assigned to an interface is configured by **switchport port-security maximum** command. The default MAC address limit on an interface where port security is enabled is one.

Example

- These commands enable port security on Ethernet interface 7, sets the maximum number of assigned MAC addresses to 2, assigns two static MAC addresses to the interface, and clears the dynamic MAC addresses for the interface.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#switchport port-security
switch(config-if-Et7)#switchport port-security maximum 2
switch(config-if-Et7)#exit
switch(config)#mac address-table static 0034.24c2.8f11 vlan 10 interface
ethernet 7
switch(config)#mac address-table static 4464.842d.17ce vlan 10 interface
ethernet 7
switch(config)#clear mac address-table dynamic interface ethernet 7
switch(config)#show port-security
Secure Port     MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                    (Count)        (Count)       (Count)
--------------------------------------------------------------------------
     Et7              2             2             0             Shutdown
--------------------------------------------------------------------------
Total Addresses in System: 1
switch(config)#show port-security address
          Secure Mac Address Table
------------------------------------------------------------
Vlan    Mac Address       Type                      Ports   Remaining Age
                                                              (mins)
----    -----------       ----                      -----   -------------
  10    0034.24c2.8f11    SecureConfigured          Et7     N/A
  10    4464.842d.17ce    SecureConfigured          Et7     N/A
------------------------------------------------------------------------
Total Mac Addresses for this criterion: 2
switch(config)#
```

10.5.2.5 Loopback Interface Configuration Mode

Loopback ports are instantiated by entering loopback interface mode for the desired port number. Loopback interface mode also provides access to loopback configuration commands. that configure. Previously instantiated ports are edited by entering loopback interface mode for the specified port.

The **interface loopback** command places the switch in interface-loopback configuration mode for the specified interface. The command creates a loopback interface for each one it specifies that was not previously created.

The switch provides loopback configuration mode for creating loopback interfaces and modifying their operating parameters:

Example

- These commands instantiate loopback interface 2 and assigns it IP address 10.1.1.42/24.

```
switch(config)#interface loopback 2
switch(config-if-Lo2)#ip address 10.1.1.42
switch(config-if-Lo2)#show active
interface Loopback2
   ip address 10.1.1.42/24
switch(config-if-Lo2)#s
```

10.5.2.6 MTU Configuration

The **mtu** command configures the IPv4 and IPv6 Maximum Transmission Unit (MTU) size for the configuration mode interface. The switch fragments IP packets that are larger than the MTU value for the outbound interface. An interface's MTU value is displayed with the **show interface** command. The command is valid for all routable interfaces.

Example

- This command sets the MTU size of 1492 bytes on VLAN interface 20.

```
switch(config-if-Vl20)#mtu 1492
switch(config-if-Vl20)#show active
interface Vlan20
   mtu 9212
   ip address 172.17.254.2/30
switch(config-if-Vl20)#
```

This command displays status and configuration information for VLAN 20.

```
switch(config)#show interface vlan 20
Vlan20 is up, line protocol is up (connected)
  Hardware is Vlan, address is 001e.4894.53a9 (bia 001e.4894.53a9)
  Description: ar.mg.mlag.peer
  Internet address is 168.31.21.4/28
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  MTU 1492 bytes
  Up 18 days, 2 hours, 54 minutes, 41 seconds
switch(config)#
```

10.5.3 Control Plane ACLs

Applying an ACL to the Control Plane

The control plane supports routing and management functions, handling packets that are addressed to the switch without regard to any switch interface.

To apply an IP ACL to the control plane, enter **ip access-group (Control Plane mode)** in control-plane mode. The **control-plane** command places the switch in control-plane mode.

Section 13.2 describes access control lists.

Example

- These commands place the switch in Control Plane mode and assigns ***CP-Test1*** to the control plane.

```
Switch(config)#control-plane
Switch(config-cp)#ip access-group CP-Test1 in
Switch(config-cp)#
```

10.5.4 Data Plane Forwarding Mode

Arista data planes support two packet forwarding modes: store-and-forward and cut-through.

- Store and forward: the switch accumulates entire packets before forwarding them.
- Cut through: the switch begins forwarding frames before their reception is complete.

Packet forwarding mode availability varies by switch platform:

- FM4000: cut-through mode only
- Petra: store and forward mode only
- Trident: both modes are available.

The **switch forwarding-mode** command specifies the forwarding mode of the switch's data plane. This command is only available on Trident platform switches. The forwarding mode is ***cut through*** on FM4000 platform switches and ***store-and-control*** on Petra platform switches.

Example

- This command changes the forwarding mode to ***store-and-forward***.

```
switch(config)#switch forwarding-mode store-and-forward
switch(config)#
```

The **show switch forwarding-mode** command displays the switch's forwarding mode.

Example

- This command displays the switch's forwarding mode.

```
switch(config)#show switch forwarding-mode
Current switching mode:    store and forward
Available switching modes: cut through, store and forward
```

10.5.5 Configuring VRFs

10.5.5.1 Main VRF

The default VRF on Arista switches is called "main." It is created automatically and cannot be renamed or configured. Some configuration options accept "main" as a VRF input.

10.5.5.2 User Defined VRFs

A single user defined VRF can be configured on Arista switches. The primary use of the user defined VRF is as a management VRF to segregate management traffic.

The user defined VRF is created with the **vrf definition** command. A user defined VRF that exists on the switch must be removed with the **no vrf definition** command before creating a new VRF. After its creation, a VRF is activated by assigning it a route distinguisher with the **rd (VRF configuration mode)** command.

Example

- These commands create a VRF named "purple," place the switch in VRF configuration mode for that VRF, and specify a route distinguisher for the VRF identifying the administrator as AS 530 and assigning 12 as its local number.

```
switch(config)#vrf definition purple
switch(config-vrf-purple)#rd 530:12
switch(config-vrf-purple)#
```

To add interfaces to the user defined VRF, enter the configuration mode for the interface and use the **vrf forwarding** command. VLAN interfaces and management interfaces can be added to the user defined VRF.

Example

- These commands add VLAN 20 to the VRF named "purple."

```
switch(config)#interface VLAN 20
switch(config-if-Vl20)#vrf forwarding purple
switch(config-if-Vl20)#
```

The **show vrf** command shows information about user defined VRFs on the switch.

Example

- This command displays information for the VRF named "purple".

```
switch>show vrf purple
   Vrf         RD              Protocols      State         Interfaces
---------- -------------- -------------- -------------- ------------
   purple      64496:237      ipv4           no routing    Vlan42, Vlan43

switch>
```

10.5.5.3 Context Active VRF

The context-active VRF specifies the default VRF that VRF-context aware commands use when displaying or refreshing routing table data from a specified VRF.

VRF-context aware commands include:

- **clear arp-cache**
- **show ip**
- **show ip arp**
- **show ip route**
- **show ip route gateway**
- **show ip route host**
- **ping**
- **tracerout**
- **telnet**
- **tcpdump**

The **routing-context vrf** command specifies the context active VRF.

Example

- This command specifies ***magenta*** as the context-active VRF.

```
switch#routing-context vrf magenta
switch#show routing-context vrf
Current VRF routing-context is magenta
```

The **show routing-context vrf** command displays the context active VRF.

Example

- This command displays the context-active VRF.

```
switch>show routing-context vrf
Current VRF routing-context is magenta
switch>
```

10.6 Data Transfer Command Descriptions

This section contains descriptions of the CLI commands that this chapter references.

Control Plane and Data Plane Commands

- control-plane . . . . . Page 343
- ip access-group (Control Plane mode) . . . . . Page 347
- switch forwarding-mode. . . . . Page 375
- show switch forwarding-mode . . . . . Page 370

MAC Address Table Commands

- clear mac address-table dynamic. . . . . Page 342
- mac address-table aging-time. . . . . Page 349
- mac address-table static. . . . . Page 350
- switchport mac address learning. . . . . Page 377
- show mac address-table . . . . . Page 360
- show mac address-table aging time . . . . . Page 362
- show mac address-table count . . . . . Page 363

Distributed Routing Commands

- exclude . . . . . Page 345
- ip distributed-hosts . . . . . Page 348
- threshold fecs . . . . . Page 380
- vlan (distributed-hosts configuration) . . . . . Page 382

Port Configuration Commands

- description . . . . . Page 344
- interface loopback . . . . . Page 346
- switchport . . . . . Page 376
- mtu . . . . . Page 355
- show interfaces description . . . . . Page 359

Port Mirroring Commands

- monitor session destination . . . . . Page 353
- monitor session source . . . . . Page 354
- no monitor session . . . . . Page 356
- show monitor session . . . . . Page 364

Storm Control Commands

- storm-control. . . . . Page 373
- show storm-control . . . . . Page 369

Port Security Commands

- switchport port-security . . . . . Page 378
- switchport port-security maximum. . . . . Page 379
- show port-security . . . . . Page 365
- show port-security address. . . . . Page 366
- show port-security interface. . . . . Page 367

Tracking Commands

- track . . . . . Page 381
- show track . . . . . Page 371

VRF Commands

- rd (VRF configuration mode) . . . . . Page 357
- routing-context vrf . . . . . Page 358
- vrf definition . . . . . Page 383
- vrf forwarding. . . . . Page 384
- show routing-context vrf. . . . . Page 368
- show vrf . . . . . Page 372

clear mac address-table dynamic

The **clear mac address-table dynamic** command removes specified dynamic entries from the MAC address table. Entries are identified by their VLAN and layer 2 (Ethernet or port channel) interface.

- To remove a specific entry, include its VLAN and interface in the command.
- To remove all dynamic entries for a VLAN, do not specify an interface.
- To remove all dynamic entries for an interface, do not specify a VLAN.
- To remove all dynamic entries, do not specify a VLAN or an interface.

Command Mode

Privileged EXEC

Command Syntax

```
clear mac address-table dynamic [VLANS] [INTERFACE]
```

Parameters

- ***VLANS*** VLAN for which command clears table entries. Options include:
  - <no parameter> all VLANs.
  - **vlan** *v_num* VLAN specified by *v_num*.
- ***INTERFACE*** Interface for which command clears table entries. Options include:
  - <no parameter> all Ethernet and port channel interfaces.
  - **interface ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **interface port-channel** *p_range* port channel interfaces specified by *p_range*.

  Valid *e_range* and *p_range* formats include number, range, or comma-delimited list of numbers and ranges.

Example

- This command clears all dynamic mac address table entries for port channel 5 on VLAN 34.

```
switch#clear mac address-table dynamic vlan 34 interface port-channel 5
switch#
```

control-plane

The **control-plane** command places the switch in control-plane configuration mode. Control-plane mode is used for assigning an ACL (access control list) to the control plane.

Control-plane configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. Exiting control-plane configuration mode does not affect the configuration.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
control-plane
```

Commands Available in control-plane configuration mode

- **ip access-group (Control Plane mode)**

Examples

- This command places the switch in control plane mode.

```
switch(config)#control-plane
switch(config-cp)
```

- This command assigns the ***control-plane-2*** ACL to the control plane.

```
switch(config-cp)#ip access-group control-plane-2
switch(config-cp)
```

- This command exits control plane mode.

```
switch(config-cp)#exit
switch(config)
```

description

The **description** command adds comment text for the configuration mode interface. The text provides information about the interface and has no effect on interface functions. The **show interfaces description** command displays interface description text.

The **no description** command removes the description text for the configuration mode interface from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
description label_text
no description
default description
```

Parameters

- *label_text* character string assigned to description attribute.

Examples

- These commands add description text to ethernet interface 23, then displays the text through a **show interfaces description** command.

```
switch(config)#interface ethernet 23
switch(config-if-Et23)#description external line
switch(config-if-Et23)#show interfaces ethernet 23 description
Interface                      Status         Protocol Description
Et23                           up             up       external line
```

exclude

The **exclude** command specifies an ACL which lists IPv4 hosts that are to be excluded from FEC distribution across the ASICs when the distributed routing feature is enabled. Routes to the IPv4 hosts listed in the specified ACL will be included in the FEC tables of all ASICs, assuring that traffic to those hosts will be single-hopped rather than internally recirculated. By default, the assignment of incoming IPv4 hosts to 1-hop or 2-hop forwarding depends on the order of their ARP resolution. The **exclude** command allows you to specify hosts for which you prefer 1-hop forwarding.

To create the ACL for IPv4 host exclusion, use the **ip access-list** command.

The **no exclude** and **default exclude** commands discontinue use of the ACL by removing the **exclude** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
exclude hosts ACL_name
no exlude hosts
default exclude hosts
```

Parameters

- *ACL_name* access control list that specifies IPv4 hosts for exclusion.

Example

- These commands create an ACL called 13-priority-acl listing two IPv4 hosts which require single-hop forwarding, and apply that ACL to exclude those hosts from FEC distribution.

```
switch(config)#ip access-list standard l3-priority-acl
switch(config-std-acl-l3-priority-acl)#permit 1.1.1.1/32
switch(config-std-acl-l3-priority-acl)#permit 2.2.0.0/16
switch(config-std-acl-l3-priority-acl)#end
switch(config)#ip distributed-hosts
switch(config-distributed-hosts)#exclude hosts l3-priority-acl
switch(config-distributed-hosts)#end
switch(config)#
```

interface loopback

The **interface loopback** command places the switch in loopback-interface configuration mode for the specified interfaces. The command creates loopback interfaces for previously unconfigured interfaces.

The command can specify a single interface or multiple interfaces:

- Single interface: Command creates an interface if it specifies one that was not previously created.
- Multiple interfaces: Command is valid only if all specified interfaces were previously created.

The **no interface loopback** command removes the specified interfaces from ***running-config***, including all interface configuration statements. The **default interface loopback** command removes all configuration statements for the specified loopback interface without deleting the loopback interface from ***running-config***.

The following commands are available in loopback configuration mode:

- description
- exit
- ip address
- ip proxy-arp
- ipv6 address
- ipv6 enable
- load interval
- logging event
- mtu
- shutdown (Interfaces)
- snmp trap

Command Mode

Global Configuration

Command Syntax

```
interface loopback l_range
no interface loopback l_port
```

Parameters

- *l_port* Numerical label of loopback interface. Value ranges from 0 to 1000.
- *l_range* Numerical label of loopback interfaces. Formats include number (0 through 1000), number range, or comma-delimited list of numbers and ranges.

Examples

- This command enters interface configuration mode for loopback interfaces 1 through 5.

```
switch(config)#interface loopback 1-5
switch(config-if-Lo1-5)#
```

- This command creates interface 23 and enters interface configuration mode:

```
switch(config)#interface loopback 23
switch(config-if-Lo23)#
```

- This command removes loopback interfaces 5 through 7 from ***running-config***.

```
switch(config)#no interface loopback 5-7
switch(config)#
```

ip access-group (Control Plane mode)

The **ip access-group** command applies an IPv4 or standard IPv4 access control list (ACL) to the control plane.

The **no ip access-group** and **default ip access-group** commands remove the corresponding **ip access-group** command from ***running-config***.

Command Mode

Control-Plane

Command Syntax

```
ip access-group list_name [VRF_INSTANCE] DIRECTION
no ip access-group [list_name] [VRF_INSTANCE] DIRECTION
default ip access-group [list_name] [VRF_INSTANCE] DIRECTION
```

Parameters

- *list_name* name of ACL assigned to interface.
- ***VRF_INSTANCE*** specifies the VRF instance being modified.
  - — <no parameter> changes are made to the main VRF.
  - — **vrf** *vrf_name* changes are made to the specified user defined VRF.
- ***DIRECTION*** transmission direction of packets, relative to interface. Valid options include:
  - — **in** inbound packets.

Example

- These commands applies the IPv4 ACL named ***test2*** to the control plane.

```
switch(config)#control-plane
switch(config-cp)#ip access-group test2 in
switch(config-cp)#
```

ip distributed-hosts

The **ip distributed-hosts** command enables the distributed routing feature and enters distributed-hosts configuration mode. Distributed-hosts configuration mode is a group change mode; changes made while in the mode are saved by leaving the mode through the exit command or by entering another configuration mode.

The total number of connected IPv4 hosts that can be accommodated on a switch is limited by the size of the FEC and LPM tables. By default, the distributed routing feature is disabled, and complete tables are maintained on every ASIC. The distributed routing feature distributes FEC and LPM tables across the ASICs on all line cards, making it possible for the switch to accomodate up to 16,000 IPv4 hosts (the size of the MAC address table) when MAC address learning is needed, or up to 24,000 directly attached IPv4 hosts if the switch is using only routed ports. When a packet is received on an ASIC which does not have the needed next-hop information to forward it, the packet is sent to the recirculation port of an ASIC which has that information.

By default, when the distributed routing feature is enabled 7250 FECs per ASIC are set aside for IPv4 host distribution. To configure that value, use the **threshold fecs** command.

The distributed routing feature designates all incoming IPv4 hosts as single-hop (including their routes in the FEC tables of all ASICs) as long as there is space to do so, then downgrades non-priority hosts to two-hop status when room is needed for additional hosts. To specify which hosts should be guaranteed single-hop status, use the **exclude** command.

The **no ip distributed-hosts** and **default ip distributed-hosts** commands disable the distributed routing feature by removing the **ip distributed-hosts** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip distributed-hosts
```

Commands Available in Distributed-Hosts Configuration Mode

- **exclude**
- **threshold fecs**
- **vlan (distributed-hosts configuration)**

Guidelines

The distributed routing feature is available only on modular Petra platforms.

The feature should be enabled before any hosts are added; otherwise you must run the **clear arp** command, which can result in extremely high CPU usage while the tables are resolving.

For the feature to function, it must be configured with an internal VLAN in the active state. It is recommended that you disable autostate on the distributed-hosts VLAN.

Examples

- This command enables the distributed routing feature and enters distributed-hosts configuration mode.

```
Switch(config)#ip distributed-hosts
Switch(config-distributed-hosts)#
```

mac address-table aging-time

The **mac address-table aging-time** command configures the aging time for MAC address table dynamic entries. Aging time defines the period an entry is in the table, as measured from the most recent reception of a frame on the entry's VLAN from the specified MAC address. The switch removes entries when their presence in the MAC address table exceeds the aging time.

Aging time ranges from 10 to 1,000,000 seconds with a default of 300 seconds (five minutes).

The **no mac address-table aging-time** and **default mac address-table aging-time** commands reset the aging time to its default by removing the **mac address-table aging-time** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
mac-address-table aging-time period
no mac-address-table aging-time
default mac-address-table aging-time
```

Parameters

- *period* MAC address table aging time. Default is 300 seconds. Options include:
  - **0** disables deletion of table entries on the basis of aging time.
  - **10** through **1000000** (one million) aging time period (seconds).

Examples

- This command sets the MAC address table aging time to two minutes (120 seconds).

```
switch(config)#mac address-table aging-time 120
switch(config)#
```

mac address-table static

The **mac address-table static** command adds a static entry to the MAC address table. Each table entry references a MAC address, a VLAN, and a list of layer 2 (Ethernet or port channel) ports. The table supports three entry types: unicast drop, unicast, and multicast.

- A drop entry does not include a port.
- A unicast entry includes one port.
- A multicast entry includes at least one port.

  Packets with a MAC address (source or destination) and VLAN specified by a drop entry are dropped. Drop entries are valid for only unicast MAC addresses.

The command replaces existing dynamic or static table entries with the same VLAN-MAC address. Static entries are not removed by aging (**mac address-table aging-time**). Static MAC entries for mirror destinations or LAG members are typically avoided.

The most significant byte of a MAC address distinguishes it as a unicast or multicast address:

- Unicast: most significant byte is an even number. Examples: 0200.0000.0000 1400.0000.0000
- Multicast: most significant byte is an odd number. Examples: 0300.0000.0000 2500.0000.0000

The **no mac address-table static** and **default mac address-table static** commands remove the corresponding **mac address-table static** command from ***running-config*** and the MAC address table entry.

Command Mode

Global Configuration

Command Syntax

```
mac address-table static mac_address vlan v_num PORT_LIST
no mac address-table static mac_address vlan v_num [PORT_LIST]
default mac address-table static mac_address vlan v_num [PORT_LIST]
```

Parameters

- *mac_address* Table entry's MAC address (dotted hex notation – H.H.H).
- *v_num* Table entry's VLAN.
- ***DESTINATION*** Table entry's port list.

  For multicast MAC address entries, the command may contain multiple ports, listed in any order. The CLI accepts only one interface for unicast entries.

  — **drop** creates drop entry in table. Valid only for unicast addresses.
  — **interface ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  — **interface port-channel** *p_range* Port channel interfaces specified by *p_range*.
  — <no parameter> Valid for **no** and **default** commands that remove multiple table entries.

  e_range and *p_range* formats include number, range, or comma-delimited list of numbers and ranges.

Examples

- This command adds the static entry for unicast MAC address 0012.3694.03ec to the MAC address table.

```
switch(config)#mac address-table static 0012.3694.03ec vlan 3 interface Ethernet 7
switch(config)#show mac address-table static
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
   3    0012.3694.03ec    STATIC      Et7
Total Mac Addresses for this criterion: 1

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0

switch(config)#
```

- These commands adds the static drop entry for MAC address 0012.3694.03ec to the MAC address table, then displays the entry in the MAC address table.

```
switch(config)#mac address-table static 0012.3694.03ec vlan 3 drop
switch(config)#show mac address-table static
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
   1    0012.3694.03ec    STATIC
Total Mac Addresses for this criterion: 1

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0

switch(config)#
```

- This command adds the static entry for the multicast MAC address 0112.3057.8423 to the MAC address table.

```
switch(config)#mac address-table static 0112.3057.8423 vlan 4 interface
port-channel 10 port-channel 12
switch(config)#show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
Total Mac Addresses for this criterion: 0

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
   4    0112.3057.8423    STATIC      Po10 Po12
Total Mac Addresses for this criterion: 1
switch(config)#
```

monitor session destination

The **monitor session destination** command configures an Ethernet interface as the destination port of a specified port mirroring session. The **monitor session source** command configures the source port of the mirroring session.

An interface cannot be used in more than one mirror session and cannot be simultaneously a source and destination. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from one direction.

The **no monitor session destination** and **default monitor session destination** commands removes the mirror session destination assignment by deleting the corresponding **monitor session destination** command from ***running-config***. The **no monitor session** removes entire the mirror session.

Command Mode

Global Configuration

Command Syntax

```
monitor session session_name source INT_NAME
no monitor session session_name source INT_NAME
default monitor session session_name source INT_NAME
```

Parameters

- *session_name* Label assigned to port mirroring session.
- ***INT_NAME*** Destination interface for the mirroing session
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.

Examples

- This command configures Ethernet interface 8 as the destination port for the ***redirect_1*** mirroring session.

```
switch(config)#monitor session redirect_1 destination ethernet 8
switch(config)#
```

monitor session source

The **monitor session source** command configures an Ethernet or port channel interface as the source port of a specified port mirroring session. The **monitor session destination** command configures the destination port of the mirroring session.

An interface cannot be used in a more than one mirror session and cannot be simultaneously a source and destination. By default, mirror sessions duplicate ingress and egress traffic but are configurable to mirror traffic from one direction.

The **no monitor session source** and **default monitor session source** commands removes the mirror session source assignment by deleting the corresponding **monitor session source** command from ***running-config***. The **no monitor session** removes entire the mirror session.

Command Mode

Global Configuration

Command Syntax

```
monitor session session_name source INT_NAME DIRECTION
no monitor session session_name source INT_NAME DIRECTION
default monitor session session_name source INT_NAME DIRECTION
```

Parameters

- *session_name* Label assigned to port mirroring session.
- ***INT_NAME*** Source interface for the mirroing session.
  - — **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - — **port-channel** *p_range* Port channel Interfaces specified by *p_range*.
- ***DIRECTION*** transmission direction of traffic to be mirrored.
  - — <no parameter> mirrors transmitted and received traffic.
  - — **both** mirrors transmitted and received traffic.
  - — **rx** mirrors received traffic.
  - — **tx** mirrors transmitted traffic.

Examples

- This command configures Ethernet interface 7 as the source port for the ***redirect_1*** mirroring session.

```
switch(config)#monitor session redirect_1 source ethernet 7
switch(config)#
```

mtu

The **mtu** command configures the IPv4 and IPv6 Maximum Transmission Unit (MTU) size for the configuration mode interface. The switch fragments IP packets that are larger than the MTU value for the outbound interface. The default MTU is 9212 bytes. An interface's MTU value is displayed with the show interface command. This command is valid for all routable interfaces.

The **no mtu** command restores the interface's MTU to the default value by removing the corresponding mtu command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

mtu *bytes*
no mtu

Parameters

- *bytes* MTU size (bytes). Values range from 68 to 65535.

Examples

- This command sets the MTU size of 1492 bytes on VLAN interface 20.

```
switch(config)#interface vlan 20
switch(config-if-Vl20)#mtu 1492
switch(config-if-Vl20)#
```

no monitor session

The **no monitor session** command removes the specified monitor session from the switch by deleting all corresponding monitor commands from ***running-config***. Commands that remove or alter individual commands within a session configuration are described in the **monitor session destination** and **monitor session source** commands.

Command Mode

Global Configuration

Command Syntax

no monitor session *session_name*

Parameters

- *session_name* Label assigned to port mirroring session.

Example

- This command displays the configuration of the ***redirect*** mirroring session, deletes the session, then confirms that the session was removed.

```
switch(config)#show monitor session redirect_1

Session redirect_1
------------------------

Source Ports

  Both:        Et7

Destination Port: Et8
switch(config)#no monitor session redirect_1
switch(config)#show monitor session redirect_1
Session not created

switch(config)#
```

rd (VRF configuration mode)

The **rd** command adds a route distinguisher (RD) to the configuration mode VRF. RDs internally identify routes belonging to a VRF to distinguish overlapping or duplicate IP address ranges. This allows the creation of distinct routes to the same IP address for different VPNs. The RD is a 64-bit number made up of an AS number or IPv4 address followed by a user-selected ID number.

Once an RD has been assigned to a VRF it cannot be changed. If the RD must be changed, remove the VRF using the **no default** form of the **vrf definition** command, then create it again.

Command Mode

VRF Configuration

Command Syntax

rd *admin_ID*:*local_assignment*

Parameters

- *admin_ID* An AS number or globally assigned IPv4 address identifying the entity assigning the RD. This should be an IANA-assigned identifying number.
- *local_assignment* A locally assigned number distinguishing the VRF. Values range from 0-65535 if the *admin_ID* is an IPv4 address, or from 0-4,294,967,295 if the *admin_ID* is an AS number. If the *admin_ID* is an AS number, the *local_assignment* can also be entered in the form of an IPv4 address.

Examples

- These commands identify the administrator of the VRF called “purple” as AS 530 and assign 12 as its local number.

```
switch(config)#vrf definition purple
switch(config-vrf-purple)#rd 530:12
switch(config-vrf-purple)#
```

routing-context vrf

The **routing-context vrf** command specifies the context-active VRF. The context-active VRF determines the default VRF that VRF-context aware commands use when displaying routing table data from a specified VRF.

Command Mode

Privileged EXEC

Command Syntax

```
routing-context vrf [VRF_ID]
```

Parameters

- ***VRF_ID*** Name of VRF assigned as the current VRF scope. Options include:
  - *vrf_name* Name of user defined VRF.
  - **main** Name of system-default VRF.

Guidelines

VRF-context aware commands include:

- **clear arp-cache**
- **show ip**
- **show ip arp**
- **show ip route**
- **show ip route gateway**
- **show ip route host**
- **ping**
- **tracerout**
- **telnet**
- **tcpdump**

Related Commands

- **show routing-context vrf** displays the context active VRF.

Example

- These commands specify ***magenta*** as the context-active VRF, then displays the context-active VRF.

```
switch#routing-context vrf magenta
switch#show routing-context vrf
Current VRF routing-context is magenta
switch#
```

show interfaces description

The **show interfaces description** command displays the status and description text of the specified interfaces. The **description** command configures an interface's description parameter.

Command Mode

EXEC Configuration

Command Syntax

show interfaces [***INT_NAME***] **description**

Parameters

- *INT_NAME* Interface type and labels. Options include:
  - <no parameter> all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.

  Valid *e_range*, *l_range*, *m_range*, *p_range*, and *v_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command displays description text and status of ethernet interfaces 1-10.

```
switch>show interfaces ethernet 1-10 description
Interface                      Status         Protocol Description
Et1                            up             up       ctar_01
Et2                            up             up       ctar_02
Et3                            up             up       ctar_03
Et4                            up             up       fobd_01
Et5                            up             up       fobd_02
Et6                            up             up       yzrq_01
Et7                            up             up       yzrq_02
Et8                            down           down     yzrq_03
Et9                            up             up       yzrq_04
Et10                           up             up       yzrq_05
switch>
```

show mac address-table

The **show mac-address-table** command displays the specified MAC address table entries.

Command Mode

EXEC

Command Syntax

```
show mac address-table [ENTRY_TYPE] [MAC_ADDR] [INTERFACE] [VLANS]
```

Parameters

- *ENTRY_TYPE* command filters display by entry type. Entry types include mlag-peer, dynamic, static, unicast, multicast entries, and configured.
  - <no parameter> all table entries.
  - **configured** static entries; includes unconfigured VLAN entries.
  - **dynamic** entries learned by the switch.
  - **multicast** entries with multicast MAC address.
  - **static** entries entered by CLI commands and include a configured VLAN.
  - **unicast** entries with unicast MAC address.
  - **mlag-peer** all MLAG peer entries.
  - **[mlag-peer] configured** static entries on MLAG peer; includes unconfigured VLAN entries.
  - **[mlag-peer] dynamic** entries learned on MLAG peer.
  - **[mlag-peer] static** MLAG entries entered by CLI commands and include a configured VLAN.
  - **[mlag-peer] unicast** MLAG entries with unicast MAC address.
- *MAC_ADDR* command uses MAC address to filter displayed entries.
  - <no parameter> all MAC addresses table entries.
  - **address** *mac_address* displays entries with specified address (dotted hex notation – H.H.H).
- *INTERFACE* command filters display by port list. When parameter lists multiple interfaces, command displays all entries containing at least one listed interface.
  - <no parameter> all Ethernet and port channel interfaces.
  - **ethernet** *e_range* Ethernet interfaces specified by *e_range*.
  - **port-channel** *p_range* Port channel interfaces specified by *p_range*.
- *VLANS* command filters display by VLAN.
  - <no parameter> all VLANs.
  - **vlan** *v_num* VLANs specified by *v_num*.

Example

- This command displays the MAC address table.

```
switch>show mac address-table
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
 101    001c.8224.36d7    DYNAMIC     Po2        1       9 days, 15:57:28 ago
 102    001c.8220.1319    STATIC      Po1
 102    001c.8229.a0f3    DYNAMIC     Po1        1       0:05:05 ago
 661    001c.8220.1319    STATIC      Po1
 661    001c.822f.6b22    DYNAMIC     Po7        1       0:20:10 ago
3000    001c.8220.1319    STATIC      Po1
3000    0050.56a8.0016    DYNAMIC     Po1        1       0:07:38 ago
3902    001c.8220.1319    STATIC      Po1
3902    001c.822b.a80e    DYNAMIC     Po4        2       9 days, 15:57:30 ago
3903    001c.8220.1319    STATIC      Po1
3903    001c.822c.3009    DYNAMIC     Po5        1       4 days, 15:13:03 ago
3908    001c.8220.1319    STATIC      Po1
3908    001c.822c.4e1d    DYNAMIC     Po1        1       0:07:26 ago
3908    001c.822c.55d9    DYNAMIC     Po1        1       0:04:33 ago
3909    001c.8220.1319    STATIC      Po1
3909    001c.822f.6a80    DYNAMIC     Po1        1       0:07:08 ago
3910    001c.730f.6a80    DYNAMIC     Et9        1       4 days, 15:13:07 ago
3911    001c.8220.1319    STATIC      Po1
3911    001c.8220.40fa    DYNAMIC     Po8        1       1:19:58 ago
3912    001c.822b.033e    DYNAMIC     Et11       1       9 days, 15:57:23 ago
3913    001c.8220.1319    STATIC      Po1
3913    001c.822b.033e    DYNAMIC     Po1        1       0:04:35 ago
3984    001c.8220.178f    DYNAMIC     Et8        1       4 days, 15:07:29 ago
3992    001c.8220.1319    STATIC      Po1
3992    001c.8221.07b9    DYNAMIC     Po6        1       4 days, 15:13:15 ago
Total Mac Addresses for this criterion: 25

          Multicast Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
Total Mac Addresses for this criterion: 0
switch>
```

show mac address-table aging time

The **show mac-address-table aging time** command displays the aging time for MAC address table dynamic entries. Aging time defines the period an entry is in the table, as measured from the most recent reception of a frame on the entry's VLAN from the specified MAC address. The switch removes entries that exceed the aging time.

Aging time ranges from 10 seconds to 1,000,000 seconds with a default of 300 seconds (five minutes).

Command Mode

EXEC

Command Syntax

```
show mac address-table aging-time
```

Examples

- This command displays the MAC address table aging time

```
switch>show mac address-table aging-time
Global Aging Time:  120
switch>
```

show mac address-table count

The **show mac-address-table count** command displays the number of entries in the MAC address table for the specified VLAN or for all VLANs.

Command Mode

EXEC

Command Syntax

show mac address-table count [***VLANS***]

Parameters

- ***VLANS*** The VLANs for which the command displays the entry count.
  - — <no parameter> all configured VLANs.
  - — **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- This command displays the number of entries on VLAN 39

```
switch>show mac address-table count vlan 39

Mac Entries for Vlan 39:
---------------------------
Dynamic Address Count            : 1
Unicast Static  Address Count    : 1
Multicast Static  Address Count  : 0
Total Mac Addresses              : 2

switch>
```

show monitor session

The **show monitor session** command displays the configuration of the specified port mirroring session. The command displays the configuration of all mirroring sessions on the switch when the session name parameter is omitted.

Command Mode

EXEC

Command Syntax

show monitor session *SESSION_NAME*

Parameters

- *SESSION_NAME* Port mirroring session identifier. Options include:
  - <no parameter> displays configuration for all sessions.
  - *label* command displays configuration of the specified session.

Example

- This command displays the mirroring configuration of the specified monitor session.

```
switch>show monitor session redirect_1

Session redirect_1
------------------------

Source Ports

  Both:        Et7

Destination Port: Et8
switch(config)>
```

show port-security

The **show port-security** command displays a summary of MAC address port securty configuration and status on each interface where switchport port security is enabled.

Command Mode
EXEC

Command Syntax
show port-security

Display Values

Each column corresponds to one physical interface. The table displays interfaces with port security displayed.

- Secure Port: Interface with switchport port-security enabled.
- MaxSecureAddr: Maximum quantity of MAC addresses that that port can process.
- CurrentAddr: Static MAC addresses assigned to the interface.
- SecurityViolation: Number of frames with unsecured addresses received by port.
- Security Action: Action triggered by a security violation.

Examples

- This command displays switchport port security configuration and status data.

```
switch>show port-security
Secure Port     MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                    (Count)        (Count)      (Count)
--------------------------------------------------------------------------------
     Et7              5             3            0            Shutdown
     Et10             1             0            0            Shutdown
--------------------------------------------------------------------------------
Total Addresses in System: 3
switch>
```

show port-security address

The **show port-security address** command display static unicast MAC addresses assigned to interfaces where switchport port security is enabled.

Command Mode
EXEC

Command Syntax
show port-security address

Example

- This command displays MAC addresses assigned to port-security protected interfaces.

```
switch>show port-security address
          Secure Mac Address Table
----------------------------------------------------------------
Vlan    Mac Address       Type                      Ports   Remaining Age
                                                              (mins)
----    -----------       ----                      -----   -------------
  10    164f.29ae.4e14    SecureConfigured          Et7     N/A
  10    164f.29ae.4f11    SecureConfigured          Et7     N/A
  10    164f.320a.3a11    SecureConfigured          Et7     N/A
----------------------------------------------------------------------------
Total Mac Addresses for this criterion: 3
switch>
```

show port-security interface

The **show port-security interface** command displays the switchport port-security status of all specified interfaces.

Command Mode
EXEC

Command Syntax
show port-security interface [***INT_NAME***]

Parameters

- *INT_NAME* Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.

  Valid *e_range*, *l_range*, *m_range*, *p_range*, and *v_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command display port-security configuration and status for the specified interfaces.

```
switch>show port-security interface ethernet 7-8
Interface                  : Ethernet7
Port Security              : Enabled
Port Status                : Secure-down
Violation Mode             : Shutdown
Maximum MAC Addresses      : 5
Aging Time                 : 5 mins
Aging Type                 : Inactivity
SecureStatic Address Aging : Disabled
Total MAC Addresses        : 3
Configured MAC Addresses   : 3
Learn/Move/Age Events      : 5
Last Source Address:Vlan   : 164f.29ae.4e14:10
Last Address Change Time   : 0:39:47 ago
Security Violation Count   : 0

Interface                  : Ethernet8
Port Security              : Disabled
Port Status                : Secure-down
Violation Mode             : Shutdown
Maximum MAC Addresses      : 1
Aging Time                 : 5 mins
Aging Type                 : Inactivity
SecureStatic Address Aging : Disabled
switch>
```

show routing-context vrf

The **show routing-context vrf** command displays the context-active VRF. The context-active VRF determines the default VRF that VRF-context aware commands use when displaying routing table data from a specified VRF.

Command Mode

EXEC

Command Syntax

```
show routing-context vrf
```

Related Commands

- **routing-context vrf** specifies the context active VRF.

Example

- This command displays the context-active VRF.

```
switch>show routing-context vrf
Current VRF routing-context is PURPLE
switch>
```

show storm-control

The **show storm-control** command displays the storm-control level and interface inbound packet capacity for the specified interface.

The configured value (**storm-control**) differs from the programmed threshold in that the hardware accounts for Interframe Gaps (IFG) based on the minimum packet size. This command displays the broadcast or multicast rate after this adjustment.

Command Mode

Privileged EXEC

Command Syntax

show storm-control [***INT_NAME***]

Parameters

- <no parameter> Command returns data for all interfaces configured for storm control.
- ***INT_NAME*** interface type and port range. Settings include:
  - **ethernet** *e_range* Ethernet interface range that *e-range* denotes. Valid *e_range* formats include a number, number range, or comma-delimited list of numbers and ranges.
  - **port-channel** *p_range* Channel group interface range that *c-range* denotes. Valid *p_range* formats include a number, number range, or comma-delimited list of numbers and ranges.

  When storm control commands exist for a port-channel and an Ethernet port that is a member of the port channel, the command for the port-channel takes precedence.

Restrictions

Storm control is available on FM4000 and Trident platform switches.

Example

- This command displays the storm control configuration for Ethernet ports 1 through 5.

```
switch#show storm-control ethernet 1-5
Port BcastEnabled BcastLevel BcastRate(Mbps) McastEnabled McastLevel McastRate(Mbps)
Et1            No        100               -           No        100               -
Et2            No        100               -           No        100               -
Et3            No        100               -          Yes         29            2976
Et4           Yes         29            2976          Yes         29            2976
Et5            No        100               -           No        100               -
```

show switch forwarding-mode

The **show switch forwarding-mode** command displays the switch's current and available forwarding plane hardware modes.

Command Mode

EXEC

Command Syntax

```
show switch forwarding-mode
```

Example

- This command changes the switch's forward mode to ***store-and-forward,*** then displays the forwarding mode.

```
switch(config)#switch forwarding-mode store-and-forward
switch(config)#show switch forwarding-mode
Current switching mode:    store and forward
Available switching modes: cut through, store and forward
```

show track

The **show track** command displays information about tracked objects configured on the switch.

Command Mode
EXEC

Command Syntax
show track [*OBJECT*] [*INFO_LEVEL*]

Parameters

- ***OBJECT*** tracked object for which information is displayed. Options include:
  - — <no parameter> displays information for all tracked objects configured on the switch.
  - — *object_name* displays information for the specified object.
- ***INFO_LEVEL*** amount of information that is displayed. Options include:
  - — <no parameter> displays complete information including object status, number of status changes, time since last change, and client process tracking the object (if any).
  - — **brief** displays brief list of all tracked objects and their current status.

Examples

- This command displays all information for tracked object ETH8.

```
switch#show track ETH8
Tracked object ETH8 is up
   Interface Ethernet8 line-protocol
      4 change, last change time was 0:36:12 ago
   Tracked by:
      Ethernet5/1 vrrp instance 50
switch#
```

- This command displays summary information for all tracked objects.

```
switch#show track brief
Tracked object ETH2 is up
Tracked object ETH4 is down
Tracked object ETH6 is up
Tracked object ETH8 is up
switch#
```

show vrf

The **show vrf** command displays the VRF name, RD, supported protocols, state and included interfaces for the specified VRF or for all VRFs on the switch.

Command Mode
EXEC

Command Syntax
show vrf [*VRF_INSTANCE*]

Parameters

- *VRF_INSTANCE* specifies the VRF instance to display.
  - — <no parameter> information is displayed for the main VRFs.
  - — **vrf** *vrf_name* information is displayed for the specified user-defined VRF.

Example

- This command displays information for the VRF named "purple."

```
switch>show vrf purple
   Vrf          RD              Protocols       State         Interfaces
------------ --------------- --------------- ---------------- --------------
   purple       64496:237       ipv4            no routing    Vlan42, Vlan43

switch>
```

storm-control

The **storm-control** command configures and enables storm control on the configuration mode physical interface. The command provides three mode options:

- **storm-control all** broadcast and multicast inbound packet control.
- **storm-control broadcast** broadcast inbound packet control.
- **storm-control multicast** multicast inbound packet control.

An interface configuration can contain three storm-control statements, one with each mode setting. The **storm-control all** threshold overrides broadcast and multicast thresholds.

When storm control is enabled, the switch monitors inbound traffic levels over one second intervals and compares the traffic level with a specified threshold. The threshold is a percentage of the total available port bandwidth and is configurable on each interface for each transmission mode.

The **no storm-control** and **default storm-control** commands remove the corresponding **storm-control** statement from ***running-config***, disabling storm control for the specified transmission type on the configuration mode interface.

Command Mode

Interface Ethernet Configuration
Interface Port Channel Configuration

Command Syntax

```
storm-control MODE level threshold
no storm-control mode
default storm-control mode
```

Parameters

- *MODE* packet transmission type. Options include:
  - **all**
  - **broadcast**
  - **multicast**
- *threshold* Inbound packet level that triggers storm control, as a percentage of port capacity. Value ranges from 1 to 100. Storm control is suppressed by a level of 100.

  The configured value differs from the programmed threshold in that the hardware accounts for Interframe Gaps (IFG) based on the minimum packet size. The **show storm-control** command displays the broadcast or multicast rate after this adjustment.

Restrictions

Storm control is available on FM4000 and Trident platform switches.

Example

- These commands enable multicast and broadcast storm control on Ethernet interface 3 and sets the multicast threshold at 65% and broadcast threshold at 50%. During each one second interval, the interface drops all inbound multicast traffic in excess of 65% of port capacity and all inbound broadcast traffic in excess of 50%.

```
switch(config)#interface ethernet 20
switch(config-if-Et20)#storm-control multicast level 65
switch(config-if-Et20)#storm-control broadcast level 50
switch(config-if-Et20)#show active
interface Ethernet20
   storm-control broadcast level 50
   storm-control multicast level 65
switch(config-if-Et20)#
```

switch forwarding-mode

The **switch forwarding-mode** command specifies the mode of the switch's forwarding plane hardware. The default forwarding mode is ***cut through***.

To add an interface to the VRF once it is created, use the **vrf forwarding** command.

The **no switch forwarding-mode** and **default switch forwarding-mode** commands restore the default forwarding mode by removing the **switch forwarding-mode** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
switch forwarding-mode [MODE_SETTING]
no switch forwarding-mode
default switch forwarding-mode
```

Parameters

- *MODE_SETTING* Specifies the switch's forwarding plane hardware mode. Options include:
  - **cut-through** the switch begins forwarding frames before their reception is complete..
  - **store-and-forward** the switch accumulates entire packets before forwarding them.

Restrictions

This command is only available on Trident platform switches. The forwarding plane mode is ***cut through*** on all FM4000 switches and ***store-and-control*** on Petra platform switches.

Examples

- This command changes the forwarding mode to ***store-and-forward***.

```
switch(config)#switch forwarding-mode store-and-forward
switch(config)#
```

switchport

The **switchport** command places the configuration mode interface in ***switched port*** (Layer 2) mode. Switched ports are configurable as members of one or more VLANs through other switchport commands. Switched ports ignore all IP level configuration commands, including IP address assignments.

The **no switchport** command places the configuration mode interface in ***routed port*** (Layer 3) mode. Routed ports are not members of any VLANs and do not switch or bridge packets. All IP level configuration commands, including IP address assignments, apply directly to the routed port interface.

By default, Ethernet and Port Channel interfaces are in switched port mode. The **default switchport** command also places the configuration mode interface in switched port mode by removing the corresponding **no switchport** command from ***running-config***.

These commands only toggle the interface between switched and routed modes. They have no effect on other configuration states.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
switchport
no switchport
default switchport
```

Guidelines

When an interface is configured as a routed port, the switch transparently allocates an internal VLAN whose only member is the routed interface. Internal VLANs are created in the range from 1006 to 4094. VLANs that are allocated internally for a routed interface cannot be directly created or configured. The **vlan internal allocation policy** command specifies the method that VLANs are allocated.

All IP-level configuration commands, except **autostate** and **ip virtual-router,** can be used to configure a routed interface. Any IP-level configuration changes made to a routed interface are maintained when the interface is toggled to switched port mode.

A LAG that is created with the **channel-group** command inherits the mode of the member port. A LAG created from a routed port becomes a routed LAG. IP-level configuration statements are not propagated to the LAG from its component members.

Examples

- These commands put Ethernet interface 5 in routed port mode.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#no switchport
switch(config-if-Et5)#
```

- These commands returns Ethernet interface 5 to switched port mode.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#switchport
switch(config-if-Et5)#
```

switchport mac address learning

The **switchport mac address learning** command enables MAC address learning for the configuration mode interface. MAC address learning is enabled by default on all Ethernet and port channel interfaces.

The switch maintains an MAC address table for switching frames between VLAN ports. When the switch receives a frame, it associates the MAC address of the transmitting interface with the recipient VLAN and port. When MAC address learning is enabled for the recipient port, the entry is added to the MAC address table. When MAC address learning is not enabled, the entry is not added to the table.

The **no switchport mac address learning** command disables MAC address learning for the configuration mode interface. The **switchport mac address learning** and **default switchport mac address learning** commands enable MAC address learning for the configuration mode interface by deleting the corresponding **no switchport mac address learning** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport mac address learning
no switchport mac address learning
default switchport mac address learning
```

Restriction

This command is available on Petra platform switches. On all other switches, MAC address learning cannot be disabled.

Example

- These commands disables MAC address learning for Ethernet interface 8, then displays the active configuration for the interface.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#no switchport mac address learning
switch(config-if-Et8)#show active
interface Ethernet8
   no switchport mac address learning
switch(config-if-Et8)#
```

switchport port-security

The **switchport port-security** command enables MAC address port security on the configuration mode interface. Ports with port security enabled restrict traffic to a limited number of hosts, as determined by their MAC addresses. The **switchport port-security maximum** command specifies the maximum number of MAC addresses.

The **no switchport port-security** and **default switchport port-security** commands disable port security on the configuration mode interface by removing the corresponding **switchport port-security** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport port-security
no switchport port-security
default switchport port-security
```

Examples

- These commands enable port security on ethernet interface 7.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#switchport port-security
switch(config-if-Et7)#
```

switchport port-security maximum

The **switchport port-security maximum** command specifies the maximum number of MAC addresses that can be assigned to the configuration mode interface when configured as a secure port. A secure port drops frames that are not received from MAC addresses whose inclusion in the MAC address table exceeds the port-security maximum. The **switchport port-security** command configures an interface as a secure port.

The **no switchport port-security maximum** and **default switchport port-security maximum** commands restore the maximum MAC address limit of one on the configuration mode interface by removing the corresponding **switchport port-security maximum** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport port-security maximum max_addr
no switchport port-security maximum
default switchport port-security maximum
```

Parameters

- *max_addr* maximum number of MAC addresses. Value ranges from 1 to 1000. Default value is 1.

Examples

- These commands configure a maximum number of secure MAC addresses of five for port channel interface 14.

```
switch(config)#interface port-channel 14
switch(config-if-Po14)#switchport port-security maximum 5
switch(config-if-Po14)#
```

threshold fecs

The **threshold fecs** command configures the number of FECs set aside for IPv4 host distribution (total number of unicast FECs allowed per ASIC). The recommended threshold value is 8000 minus 2*(expected number of multicast FECs).

The **no threshold fecs** and **default threshold fecs** commands reset the threshold FECs value to its default of 7250 by removing the corresponding **threshold fecs** command from ***running-config***.

Command Mode

Distributed-Hosts Configuration

Command Syntax

threshold fecs *threshold_value*

Parameters

threshold_value Number of unicast FECs allowed per ASIC. Values range from 0 to 8000. The default is 7250.

Example

- This command sets the threshold FECs value to 6,000.

```
switch(config distributed-hosts)#threshold fecs 6000
switch(config distributed-hosts)#
```

track

The **track** command creates an object whose state changes to provide information to a client process. The client process must be separately configured for object tracking to have an effect on the switch.

The **no track** and **default track** commands remove the specified tracked object by removing the corresponding **track** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
track object_name interface INTERFACE_NAME PROPERTY
no track object_name
default track object_name
```

Parameters

- *object_name* IP address of (text).
- ***INTERFACE_NAME*** Interface associated with the tracked object. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
- ***PROPERTY*** Tracked property. Options include:
  - **line-protocol** Object changes when the associated interface changes state.

Example

- This command creates a tracked object which tracks the state of the line protocol on Ethernet interface 8.

```
switch(config)#track ETH8 interface ethernet 8 line-protocol
switch(config)#
```

vlan (distributed-hosts configuration)

The **vlan (distributed-hosts configuration)** command specifies a non-default distributed-hosts VLAN. For the distributed routing feature to function, it must have a configured VLAN in the active state. It is recommended to disable autostate on the distributed-hosts VLAN so that it will remain in the active state.

The **no vlan (distributed-hosts configuration)** and **default vlan (distributed-hosts configuration)** commands reset the distributed-hosts VLAN to the default value of 4093 by removing the corresponding **vlan (distributed-hosts configuration)** command from ***running-config***.

Command Mode

Distributed-Hosts Configuration

Command Syntax

vlan *vlan_ID*

Parameters

- *vlan_ID* Non-default distributed-hosts VLAN ID. Values range from 1 to 4094. The default is 4093.

Example

- These commands configure VLAN 3165 and set is as the distributed-hosts VLAN.

```
switch(config)# vlan 3165
switch(config-vlan-3165)# exit
switch(config)# int vlan 3165
switch(config-if-Vl3165)# ip address 40.93.1.1/24
switch(config-if-Vl3165)# no autostate
switch(config-if-Vl3165)# end
switch(config)#ip distributed-hosts
switch(config-distributed-hosts)#vlan 3165
switch(config-distributed-hosts)#end
```

vrf definition

The **vrf definition** command places the switch in VRF configuration mode for the specified VRF. If the named VRF does not already exist, this command creates it. Only one non-default VRF is supported by EOS. If you attempt to create a second VRF, an error is generated.

The **no vrf definition** and **default vrf definition** commands delete the specified VRF instance by removing the corresponding **vrf definition** command from ***running-config***. This also removes all IP addresses associated with interfaces that belong to the deleted VRF.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
vrf definition vrf_name
no vrf definition vrf_name
default vrf definition vrf_name
```

Parameters

- *vrf_name* Name of VRF being created, deleted or configured. The name “main” is reserved because it designates the global default VRF.

Commands Available in VRF configuration mode

- **rd (VRF configuration mode)**

Example

- This command creates a VRF named “purple” and places the switch in VRF configuration mode for that VRF.

```
switch(config)#vrf definition purple
switch(config-vrf-purple)#
```

vrf forwarding

The **vrf forwarding** command adds the configuration mode interface to the specified VRF. You must create the VRF first, using the **vrf definition** command.

The **no vrf forwarding** and **default vrf forwarding** commands remove the configuration mode interface from the specified VRF by deleting the corresponding **vrf forwarding** command from ***running-config***.

All forms of the **vrf forwarding** command remove all IP addresses associated with the configuration mode interface.

Command Mode

Interface-Management Configuration
Interface-VLAN Configuration

Command Syntax

```
vrf forwarding vrf_name
no vrf forwarding [vrf_name]
default vrf forwarding [vrf_name]
```

Parameters

- *vrf_name* name of configured VRF.

Examples

- These commands add the configuration mode interface (VLAN 20) to the VRF named “purple”.

```
switch(config)#interface vlan 20
switch(config-if-Vl20)#vrf forwarding purple
switch(config-if-Vl20)#
```

- These commands remove the configuration mode interface from VRF “purple”.

```
switch(config)#interface vlan 20
switch(config-if-Vl20)#no vrf forwarding purple
switch(config-if-Vl20)#
```

Chapter 11

VLANs

This chapter describes Arista's VLAN implementation, including private VLANs. MAC address tables are also discussed in this chapter.

Sections in this chapter include:

- Section 11.1: Introduction
- Section 11.2: VLAN Conceptual Overview
- Section 11.3: VLAN Configuration Procedures
- Section 11.4: VLAN Configuration Commands

11.1 Introduction

Arista switches support industry standard 802.1q vlans. Arista EOS provides tools to manage and extend VLANs throughout the data center network.

11.2 VLAN Conceptual Overview

11.2.1 VLAN Definition

A virtual local area network (VLAN) is a group of devices that are configured to communicate as if they are attached to the same network regardless of their physical location. VLANs are layer 2 structures. 802.1Q is a networking standard that allows multiple bridged networks to transparently share the same physical network link.

These parameters are associated with a VLAN:

- VLAN number (1-4094): VLAN numbers uniquely identify the VLAN within a network. VLAN 1 exists by default; all other VLANs only exist after they are configured.
- VLAN name (optional): The VLAN name is a text string that describes the VLAN.
- VLAN state (***active*** or ***suspended***): The state specifies the VLAN transmission status within the switch. In the ***suspended*** state, VLAN traffic is blocked on all switch ports. The default state is ***active***.

VLANs define broadcast domains in a layer 2 network. A broadcast domain is the set of devices that can receive broadcast frames originating from any device within the set. Switches accommodating multiple broadcast domains serve as multiport bridges where each broadcast domain is a distinct virtual bridge. Traffic does not pass directly between different VLANs within a switch or between two switches.

11.2.2 VLAN Switching

Ethernet and port channel interfaces are configured as switched ports by default. Switched ports are configurable as members of one or more VLANs. Switched ports ignore all IP level configuration commands, including IP address assignments.

11.2.2.1 VLAN Trunking

Trunking is a concept where multiple VLANs extend beyond the switch through a common interface or port channel. A trunk is a point-to-point link between one or more physical interfaces and other networking devices.

A trunk group is the set of physical interfaces that comprise the trunk and the collection of VLANs whose traffic is carried on the trunk. The traffic of a VLAN that belongs to one or more trunk groups is carried only on ports that are members of trunk groups to which the VLAN belongs.

VLAN traffic is carried through Ethernet or LAG ports. A port's switchport mode defines the number of VLANs for which the port can carry traffic.

- Access ports carry traffic for one VLAN – the access VLAN. Access ports associate untagged frames with the access VLAN. Access ports drop tagged frames that are not tagged with the access VLAN.
- Trunk ports carry traffic for multiple VLANs. Tag frames specify the VLAN for which trunk ports process packets.

11.2.2.2 Q-in-Q Trunking

A Q-in-Q network is a multi-tier layer 2 VLAN network. A typical Q-in-Q network is composed of a service provider network (tier 1) where each node connects to a customer network (tier 2).

802.1ad is a networking standard that supports QinQ networks by allowing multiple 802.1Q tags in an Ethernet frame.

Each interface in a customer network is assigned to a customer-VLAN (c-VLAN). Packets in c-VLANs contain 802.1q tags that switch traffic within the network. c-VLANs access the service provider VLAN (s-VLAN) through a provider switch. Customer switch ports connect to an s-VLAN through provider switch edge ports, which are configured as dot1q ports and operate as follows:

- Inbound traffic (from customer switches): adds an s-VLAN tag, then forwards packets to the provider network.
- Outbound traffic (to customer switches): removes the s-VLAN tag, then forwards packets to the customer network.

11.2.2.3 Private VLANs

A private VLAN is a network structure that partitions a single broadcast domain into multiple subdomains. Private VLANs provide peer port isolation and can provide IP address simplification over topologies that normally allocate a separate domain (VLAN) for each defined broadcast subdomain.

A private VLAN consists of a single primary VLAN and multiple secondary VLANs.

- Primary VLAN: A primary VLAN defines the entire broadcast domain and corresponds to the basic VLAN in a topology that does not include private VLANs. Primary VLAN ports communicate with secondary VLAN ports and ports external to the private VLAN.
- Secondary VLAN: Secondary VLANs define the broadcast subdomains that comprise the domain defined by their affiliated primary VLAN. Secondary VLAN types include isolated or community:

— Isolated: Isolated VLAN ports carry unidirectional traffic from host ports to primary VLAN ports. Isolated VLAN ports filter broadcast and multicast traffic (Layer 2) from all other ports in the same isolated VLAN.

— Community: Community VLAN ports carry traffic from host ports to the primary VLAN ports and to other host ports in the same community VLAN.

Secondary VLANs do not support multicast sources when multicast routing is enabled.

VLAN interfaces for secondary VLANs can be assigned but are not functional. The status of SVIs for secondary VLANs is ***protocol line down***.

11.2.3 VLAN Routing

Each VLAN can be associated with a switch virtual interface (SVI), also called a VLAN interface. The VLAN interface functions in a routed network (layer 3) with an assigned IP subnet address. Connecting different VLANs requires layer 3 networking.

11.2.3.1 VLAN Interfaces

A switched virtual interface (SVI) is a virtual routed interface that connects to the VLAN segment on the switch. The SVI provides layer 3 processing for packets from the VLAN. An SVI can be activated only after it is connected to a VLAN. SVIs are typically configured for a VLAN to a default gateway for a subnet to facilitate traffic routing with other subnets.

In a layer 3 network, each VLAN SVI is associated with an IP subnet, with all stations in the subnet members of the VLAN. Traffic between different VLANs is routed when IP routing is enabled.

11.2.3.2 Internal VLANs

A routed port is an Ethernet or port channel interface that functions as a layer 3 interface. Routed ports do not bridge frames nor switch VLAN traffic. Routed ports have IP addresses assigned to them and packets are routed directly to and from the port.

The switch allocates an internal VLAN for an interface when it is configured as a routed port. The internal VLAN is assigned a previously unused VLAN ID. The switch prohibits the subsequent configuration of VLANs and VLAN interfaces with IDs corresponding to allocated internal VLANs.

11.3 VLAN Configuration Procedures

These sections describe basic VLAN configuration tasks.

- Section 11.3.1: Creating and Configuring VLANs
- Section 11.3.2: Configuring VLAN Switching
- Section 11.3.3: Configuring Private VLANS
- Section 11.3.4: Creating and Configuring VLAN Interfaces
- Section 11.3.5: Allocating Internal VLANs

11.3.1 Creating and Configuring VLANs

The CLI provides two methods of creating VLANs.

- Explicitly through the **vlan** command.
- Implicitly through the **switchport access vlan** command.

  The **switchport access vlan** command generates a warning message when it creates a VLAN.

To create a VLAN, use the **vlan** command in global configuration mode. Valid VLAN numbers range between 1 and 4094. To create multiple VLANs, specify a range of VLAN numbers.

To edit an existing VLAN, enter the **vlan** command with the number of the existing VLAN.

Example

- This command creates VLAN 45 and enters VLAN configuration mode for the new VLAN.

```
switch(config)#vlan 45
switch(config-vlan-45)#
```

To assign a name to a VLAN, use the **name (VLAN configuration mode)** command.

Example

- These commands assign the name Marketing to VLAN 45.

```
switch(config)#vlan 45
switch(config-vlan-45)#name Marketing
switch(config-vlan-45)#show vlan 45
VLAN Name                            Status    Ports
---- ------------------------------- --------- -------------------------
45   Marketing                       active    Et1

switch(config-vlan-45)#
```

To change a VLAN's state, use the **state** command in VLAN configuration mode.

Examples

- These commands suspend VLAN 45. VLAN traffic is blocked on all switch ports.

```
switch(config)#vlan 45
switch(config-vlan-45)#state suspend
switch(config-vlan-45)#show vlan 45
VLAN Name                            Status    Ports
---- ------------------------------- --------- -------------------------
45   Marketing                       suspended

switch(config-vlan-45)#
```

- These commands activate VLAN 45.

```
switch(config)#vlan 45
switch(config-vlan-45)#state active
switch(config-vlan-45)#show vlan 45
VLAN Name                             Status    Ports
---- -------------------------------- --------- -----------------------
45   Marketing                        active    Et1

switch(config-vlan-45)#
```

11.3.2 Configuring VLAN Switching

The following sections describe the configuring of VLAN port types.

11.3.2.1 Access Ports

Access ports carry traffic for one VLAN, as designated by a **switchport access vlan** command. Access ports associate untagged frames with the access VLAN. Tagged frames received by the interface are dropped unless they are tagged with the access VLAN.

To configure an interface group as an access port, use the **switchport mode** command.

Example

- These commands configure Ethernet interface 1 as an access port.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#switchport mode access
switch(config-if-Et1)#
```

To specify the port's access VLAN, use the **switchport access vlan** command.

Examples

- These commands configure VLAN 15 as the access VLAN for Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#switchport access vlan 15
switch(config-if-Et5)#
```

- These commands configure Ethernet interface 1 through 3 as access ports that process untagged frames as VLAN 5 traffic.

```
switch>en
switch#config
switch(config)#interface Ethernet 1-3
switch(config-if-Et1-3)#switchport mode access
switch(config-if-Et1-3)#switchport access vlan 5
switch(config-if-Et1-3)#show interfaces ethernet 1-3 vlans
Port       Untagged Tagged
Et1        None     23,25
Et2        18       -
Et3        None     14
switch(config-if-Et1-3)#
```

11.3.2.2 Trunk Ports

Trunk ports carry traffic for multiple VLANs. Messages use tag frames to specify the VLAN for which trunk ports process traffic.

- The **vlan trunk list** specifies the VLANs for which the port handles tagged frames. The port drops any packets tagged for VLANs not in the VLAN list.
- The **native vlan** is the VLAN where the port switches untagged frames.

To configure an interface group as a trunk port, use the **switchport mode** command.

Example

- These commands configure Ethernet interface 8 as a trunk port.

```
switch(config)#interface ethernet 8
switch(config-if-Et8)#switchport mode trunk
switch(config-if-Et8)#
```

To specify the port's VLAN trunk list, use the **switchport trunk allowed vlan** command.

Examples

- These commands configure VLAN 15, 20, 21, 22, 40, and 75 as the VLAN trunk list for Ethernet interface 12-16.

```
switch(config)#interface ethernet 12-16
switch(config-if-Et12-16)#switchport trunk allowed vlan 15,20-22,40,75
switch(config-if-Et12-16)#
```

- These commands add VLAN 100 through 120 to the VLAN trunk list for Ethernet interface 14.

```
switch(config)#interface ethernet 14
switch(config-if-Et14)#switchport trunk allowed vlan add 100-120
switch(config-if-Et14)#
```

To specify the port's native VLAN, use the **switchport trunk native vlan** command.

Example

- These commands configure VLAN 12 as the native VLAN trunk for Ethernet interface 10.

```
switch(config)#interface ethernet 10
switch(config-if-Et10)#switchport trunk native vlan 12
switch(config-if-Et10)#
```

By default, ports send native VLAN traffic with untagged frames. The **switchport trunk native vlan** command can also configure the port to send native VLAN traffic with tag frames.

Examples

- These commands configure Ethernet interface 10 to send native VLAN traffic as tagged.

```
switch(config)#interface ethernet 10
switch(config-if-Et10)#switchport trunk native vlan tag
switch(config-if-Et10)#
```

- These commands configure Ethernet interface 12 as a trunk with VLAN 15 as the native VLAN. The port's trunk list includes all VLANs except 201-300. The port sends all native VLAN traffic as tagged.

```
switch(config)#interface ethernet 12
switch(config-if-Et12)#switchport mode trunk
switch(config-if-Et12)#switchport trunk native vlan 15
switch(config-if-Et12)#switchport trunk native vlan tag
switch(config-if-Et12)#switchport trunk allowed vlan except 201-300
switch(config-if-Et12)#
```

11.3.2.3 Dot1q Tunnel Ports

A dot1q tunnel port is an edge port on a provider switch in a Q-in-Q network. Dot1q-tunnel ports assumes all inbound packets are untagged traffic and handles them as traffic of its access VLAN.

To configure an interface group as a dot1q tunnel port, use the **switchport mode** command.

Example

- These commands configure Ethernet interface 12 as a dot1q tunnel port.

```
switch(config)#interface ethernet 12
switch(config-if-Et12)#switchport mode dot1q-tunnel
switch(config-if-Et12)#
```

To specify the dot1q-tunnel port's access VLAN, use the **switchport access vlan** command. The port then handles all inbound traffic as untagged VLAN traffic.

Example

- These commands configure VLAN 60 as the access VLAN for Ethernet interface 12.

```
switch(config)#interface ethernet 12
switch(config-if-Et12)#switchport access vlan 60
switch(config-if-Et12)#
```

11.3.3 Configuring Private VLANS

Private VLANs are created and configured in VLAN configuration mode. Ports are associated with VLANs by switchport commands in Ethernet interface and port channel interface configuration modes.

11.3.3.1 Creating and Configuring Private VLANs

To create a primary or secondary VLAN, use the **vlan** command in global configuration mode. The procedure is identical to creating non-private VLANs. VLAN numbers allocated to secondary VLANs are not available for other switch VLANs.

Configuring a primary VLAN does not require any additional commands. To configure a secondary VLAN, use the **private-vlan** command in VLAN configuration mode. This command specifies the type of secondary VLAN and binds it to a primary VLAN.

Secondary VLANs do not support multicast sources when multicast routing is enabled.

Example

- These commands creates a private VLAN that consists of five VLANs: VLAN 25 is the primary VLAN, VLANs 30-31 are isolated VLANs, and VLANs 32-33 are community VLANs.

```
switch(config)#vlan 25
switch(config-vlan-25)#exit
switch(config)#vlan 30-31
switch(config-vlan-30-31)#private-vlan isolated primary vlan 25
switch(config-vlan-30-31)#exit
switch(config)#vlan 32-33
switch(config-vlan-32-33)#private-vlan community primary vlan 25
switch(config-vlan-32-33)#exit
switch(config)#
```

11.3.3.2 Assigning Ports to Private VLANs

Ethernet and port channel interfaces are associated with private VLANs through **switchport** commands, similar to other VLANs, as described in Section 11.3.2.

11.3.3.3 Mapping Ports to Secondary VLANs

Traffic that the primary VLAN receives on ports mapped to secondary VLANs is also received by the primary VLAN. By default, all primary VLAN ports map to the secondary VLANs. The **switchport private-vlan mapping** and **private-vlan mapping** commands specify VLAN mappings for the configuration mode interfaces.

Example

- These commands (1) configure Ethernet interface 7 as an access port for VLAN 25, which was previously configured as a primary VLAN, and (2) map the interface to VLANs 30 through 32.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#switchport mode access
switch(config-if-Et7)#switchport access vlan 25
switch(config-if-Et7)#switchport private-vlan mapping 30-32
switch(config-if-Et7)#
```

11.3.4 Creating and Configuring VLAN Interfaces

The **interface vlan** command places the switch in VLAN-interface configuration mode for modifying an SVI. An SVI provides a management address point and Layer 3 processing for packets from all VLAN ports.

Example

- This command enters VLAN-interface configuration mode for VLAN 12. The command also creates VLAN 12 interface if it was not previously created.

```
switch#config
switch(config)#interface vlan 12
switch(config-if-Vl12)#
```

11.3.5 Allocating Internal VLANs

The **vlan internal allocation policy** command specifies the VLANs that the switch allocates as internal VLANs when configuring routed ports and the order of their allocation. By default, the switch allocates VLANs in ascending order. The default allocation range is between VLAN 1006 and VLAN 4094.

The **no switchport** command converts an Ethernet or port channel interface into a routed port, disabling layer 2 switching for the interface.

Examples

- This command configures the switch to allocate internal VLANs from 1006 up.

```
switch(config)#vlan internal allocation policy ascending
switch(config)#
```

- This command configures the switch to allocate internal VLANs from 4094 down.

```
switch(config)#vlan internal allocation policy descending
switch(config)#
```

- This command configures the switch to allocate internal VLANs from 4094 down through 4000.

```
switch(config)#vlan internal allocation policy descending range 4000 4094
switch(config)#
```

11.4 VLAN Configuration Commands

This section contains descriptions of the CLI commands that this chapter references.

Global VLAN Configuration Commands

- interface vlan . . . . . Page 396
- vlan . . . . . Page 420
- vlan internal allocation policy . . . . . Page 421

VLAN Configuration Mode Commands

- name (VLAN configuration mode) . . . . . Page 397
- private-vlan . . . . . Page 398
- state . . . . . Page 412
- trunk group . . . . . Page 419

Layer 2 Interface (Ethernet and Port Channel) Configuration Commands

- switchport access vlan . . . . . Page 413
- switchport mode . . . . . Page 414
- switchport private-vlan mapping . . . . . Page 415
- switchport trunk allowed vlan . . . . . Page 416
- switchport trunk group . . . . . Page 417
- switchport trunk native vlan . . . . . Page 418

VLAN Interface Configuration Mode Commands

- autostate . . . . . Page 395
- private-vlan mapping . . . . . Page 399

Show Commands

- show dot1q-tunnel . . . . . Page 400
- show interfaces switchport . . . . . Page 401
- show interfaces switchport backup . . . . . Page 402
- show interfaces trunk . . . . . Page 403
- show interfaces vlans . . . . . Page 404
- show vlan . . . . . Page 405
- show vlan dynamic . . . . . Page 406
- show vlan internal allocation policy . . . . . Page 407
- show vlan internal usage . . . . . Page 408
- show vlan private-vlan . . . . . Page 409
- show vlan summary . . . . . Page 410
- show vlan trunk group . . . . . Page 411

autostate

Autostate is a switch feature that specifies the conditions that a VLAN interface requires to function. When autostate is ***enabled***, the following conditions are required for a VLAN interface to be in an up (status) / up (protocol) state:

- the corresponding VLAN exists and is in the active state.
- at least one layer 2 port has a link up and is in spanning-tree forwarding state on the VLAN.
- the VLAN interface exists and is not administratively down (**shutdown**).

Autostate is ***enabled*** by default. When autostate is ***disabled***, the VLAN interface is forced active.

- The **no autostate** command disables autostate on the configuration mode interface. The **no autostate** command is stored to ***running-config***.
- The **autostate** command enables the autostate function on the configuration mode VLAN SVI by removing the corresponding **no autostate** statement from ***running-config***.
- The **default autostate** command restores the autostate default state of ***enabled*** by removing the corresponding **no autostate** statement from ***running-config***.

Command Mode

Interface-VLAN Configuration

Command Syntax

```
autostate
no autostate
default autostate
```

Guidelines

Autostate should be disabled on SVIs configured as an MLAG local interface.

Examples

- These commands disable autostate on VLAN 100.

```
switch(config)#interface vlan 100
switch(config-if-Vl100)#no autostate
switch(config-if-Vl100)#
```

- These commands enable autostate on VLAN 100.

```
switch(config)#interface vlan 100
switch(config-if-Vl100)#autostate
switch(config-if-Vl100)#
```

interface vlan

The **interface vlan** command places the switch in VLAN-interface configuration mode for modifying parameters of the switch virtual interface (SVI). An SVI provides Layer 3 processing for packets from all ports associated with the VLAN. There is no physical interface for the VLAN.

When entering configuration mode to modify existing SVIs, the command can specify multiple interfaces. The command creates an SVI if the specified interface does not exist prior to issuing the command. When creating an SVI, the command can only specify a single interface.

The **no interface vlan** command deletes the specified SVI interfaces from ***running-config***. The **default interface vlan** commands remove all configuration statements for the specified SVI interfaces from ***running-config*** without deleting the interfaces.

Command Mode

Global Configuration

Command Syntax

```
interface vlan v_range
no interface vlan v_range
default interface vlan v_range
```

Parameter

- *v_range* VLAN interfaces (number, range, or comma-delimited list of numbers and ranges).

  VLAN number ranges from 1 to 4094.

Restrictions

Internal VLANs: A VLAN interface cannot be created or configured for internal VLAN IDs. The switch rejects any **interface vlan** command that specifies an internal VLAN ID.

Private VLANs: VLAN interfaces for secondary VLANs can be assigned but are not functional. The status of SVIs for secondary VLANs is ***protocol line down***.

Example

- This example creates an SVI for VLAN 12:

```
switch#config
switch(config)#interface vlan 12
switch(config-if-Vl12)#
```

name (VLAN configuration mode)

The **name** command configures the VLAN name. The name consists of up to 32 characters. The default name for VLAN 1 is ***default***. The default name for all other VLANs is VLANxxxx, where xxxx is the VLAN number. The default name for VLAN 55 is VLAN0055. The **show vlan** command displays the VLAN name.

The **name** command accepts all characters except the space.

The **no name** and **default name** commands restore the default name by removing the **name** command from ***running-config***.

Command Mode

VLAN Configuration

Command Syntax

name *label_text*
no name
default name

Parameters

- *label_text* character string assigned to name attribute. Maximum length is 32 characters. The space character is not permitted in the name string.

Examples

- These commands assign corporate_100 as the name for VLAN 25, then displays the VLAN name.

```
switch(config)#vlan 25
switch(config-vlan-25)#name corporate_100
switch(config-vlan-25)#show vlan 25
VLAN  Name                             Status    Ports
----- -------------------------------- --------- --------------------------------
25    corporate_100                    active

switch(config-vlan-25)#
```

private-vlan

The **private-vlan** command configures the configuration mode VLAN as a secondary VLAN, specifies its type, and associates it with a primary VLAN.

The **no private-vlan** and **default private-vlan** commands restores the configuration mode VLANs to their default state as primary VLANs by removing the corresponding **private-vlan** statements from *running-config*.

Command Mode

VLAN Configuration

Command Syntax

```
private-vlan [VLAN_TYPE] primary vlan v_num
no private-vlan
default private-vlan
```

Parameters

- *VLAN_TYPE* private VLAN type. Options include:
  - **community** community private VLAN.
  - **isolated** isolated private VLAN.
- *v_num* VLAN ID of primary VLAN to which the configuration mode VLAN is bound.

Restriction

Private VLANs are available on FM4000 and Trident platform switches.

Example

- These commands configure VLAN 25 as a private VLAN of type ***isolated***, binds it to VLAN 5, then displays its status as a private VLAN.

```
switch(config)#vlan 25
switch(config-vlan-25)#private-vlan isolated primary vlan 5
switch(config-vlan-25)#show vlan 25
VLAN  Name                             Status    Ports
----- -------------------------------- --------- --------------------------------
25    corporate_100                    active

switch(config-vlan-25)#show vlan private-vlan
Primary Secondary Type         Ports
------- --------- ------------ --------------------------------
5       25        isolated
switch(config-vlan-25)#
```

private-vlan mapping

The **private-vlan mapping** command maps traffic received by the configuration mode VLAN interface to a list of secondary VLANs. Command options are available to establish a new VLAN list or modify an existing list. By default, traffic to the primary VLAN interface maps to all of its secondary VLANs.

The **no private-vlan mapping** and **default private-vlan mapping** commands restore the default VLAN mapping by removing the corresponding **switchport private-vlan mapping** statement from *running-config*.

Command Mode

Interface-VLAN Configuration

Command Syntax

```
private-vlan mapping EDIT_ACTION
no private-vlan mapping
default private-vlan mapping
```

Parameters

- *EDIT_ACTION* modifications to the VLAN list.
  - *v_range* Creates VLAN list from *v_range*.
  - **add** *v_range* Adds specified VLANs to current list.
  - **except** *v_range* VLAN list contains all VLANs except those specified.

  Valid *v_range* formats include number, range, or comma-delimited list of numbers and ranges.

Restriction

Private VLANs are available on FM4000 and Trident platform switches.

Example

- These commands map VLAN interface 100 from the primary VLANs configured on the interface to VLANs 25-40.

```
switch(config)#interface vlan 100
switch(config-if-vll00)#private-vlan mapping 25-40
switch(config-if-vll00)#
```

show dot1q-tunnel

The **show dot1q-tunnel** command displays the ports that are configured in dot1q-tunnel switching mode. The **switchport mode** command configures the switching mode for the configuration mode interface.

Command Mode

EXEC

Command Syntax

```
show dot1q-tunnel [INTERFACE]
```

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.

  Valid *e_range*, *l_range*, *m_range*, *p_range*, and *v_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Restriction

Dot1q-tunnel switching mode is not available on Petra platform switches.

Example

- This command displays the ports that are configured in dot1q-tunnel switching mode.

```
switch>show dot1q-tunnel
dot1q-tunnel mode LAN Port (s)
------------------------------
Po4
Po21
Po22
switch>
```

show interfaces switchport

The **show interfaces switchport** command displays the switching configuration and operational status of the specified ports.

Command Mode

EXEC Configuration

Command Syntax

show interfaces [***INTERFACE***] **switchport**

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.

  Valid *e_range*, *l_range*, *m_range*, *p_range*, and *v_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command displays the switching status of port channel interfaces 21 and 22.

```
switch>show interface port-channel 21-22 switchport
Name: Po21
Switchport: Enabled
Administrative Mode: tunnel
Operational Mode: tunnel
Access Mode VLAN: 1 (inactive)
Trunking Native Mode VLAN: 100 (VLAN0100)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: ALL
Trunk Groups: foo

Name: Po22
Switchport: Enabled
Administrative Mode: tunnel
Operational Mode: tunnel
Access Mode VLAN: 1 (inactive)
Trunking Native Mode VLAN: 1 (inactive)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: ALL
Trunk Groups:

switch>
```

show interfaces switchport backup

The **show interfaces switchport backup** command displays interfaces that are configured as switchport backup pairs and the operational status of each interface. For each pair, the command displays the names, roles, status, and VLAN traffic of each interface.

Command Mode

EXEC

Command Syntax

```
show interfaces [INTERFACE] switchport backup
```

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.

  Valid *e_range*, *l_range*, *m_range*, *p_range*, and *v_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Display Values

- **State** Ooperational status of the interface. Values include:
  - ***Up*** Spanning tree mode is *backup*, interface status is ***up***.
  - ***Down*** Spanning tree mode is ***backup***, interface status is ***down***.
  - ***Inactive Configuration*** The spanning tree mode is not ***backup***.
- **Forwarding vlans** VLANs forward by the interface. Depends on interface operation status and prefer option specified by the **switchport backup** command.

Example

- This command displays the configured switchport primary-backup pairs.

```
switch>show interfaces switchport backup
Switch backup interface pair: Ethernet17, Ethernet18
Primary Interface: Ethernet17       State:  Up
Backup Interface:  Ethernet18       State:  Up
Ethernet17 forwarding vlans:  1-20
Ethernet18 forwarding vlans:
```

show interfaces trunk

The **show interfaces trunk** command displays configuration and status information for interfaces configured in switchport trunk mode.

Command Mode

EXEC

Command Syntax

show interfaces [***INTERFACE***] **trunk**

Parameters

- *INTERFACE* Interface type and numbers. Options include:
  - <no parameter> Display information for all interfaces.
  - **ethernet** *e_range* Ethernet interface range specified by *e_range*.
  - **loopback** *l_range* Loopback interface specified by *l_range*.
  - **management** *m_range* Management interface range specified by *m_range*.
  - **port-channel** *p_range* Port-Channel Interface range specified by *p_range*.
  - **vlan** *v_range* VLAN interface range specified by *v_range*.

  Valid *e_range*, *l_range*, *m_range*, *p_range*, and *v_range* formats include number, number range, or comma-delimited list of numbers and ranges.

Example

- This command displays the trunk status for all interfaces configured in switchport trunk mode.

```
switch>show interfaces trunk
Port            Mode            Status          Native vlan
Po1             trunk           trunking        1
Po2             trunk           trunking        1

Port            Vlans allowed
Po1             1-15
Po2             16-30

Port            Vlans allowed and active in management domain
Po1             1-10
Po2             21-30


Port            Vlans in spanning tree forwarding state
Po1             1-10
Po2             21-30

switch>
```

show interfaces vlans

The **show interfaces vlans** command displays a table that lists the VLANs that are carried by the specified interfaces. Interfaces that do not carry VLANs are not listed in the table. The table lists the untagged (native or access) and tagged VLANs for each interface.

Command Mode

EXEC

Command Syntax

show interfaces [*INT_NAME*] vlans

Parameters

- *INT_NAME* Interface type and number. Values include
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.

Example

- This command displays the VLANs carried by all L2 ports.

```
switch>show interfaces vlans
Port       Untagged Tagged
Et9        3910     -
Et11       3912     -
Et16       500      -
Et17       3908     -
Et18       3908     -
Po1        1        101-102,500,721,3000,
Po2        101      -
Po4        3902     -
Po5        3903     -
Po6        3992     -
Po7        661      -
Po8        3911     -
```

show vlan

The **show vlan** command displays the VLAN ID, name, status, and member ports of all configured VLANs. The command only displays active ports by default; by specifying ***configured-ports***, the command displays all ports that are members of a configured VLAN regardless of their activity status, including Ethernet ports that are members of a port channel.

Command Mode
EXEC

Command Syntax
show vlan [***VLAN_LIST***] [***PORT_ACTIVITY***]

Parameters

- ***VLAN_LIST*** List of VLANs displayed by command. Options include:
  - <no parameter> all VLANs.
  - *v_range* VLANs specified by *v_range*.
  - **id** *v_range* VLANs specified by *v_range*.
  - **name** *v_name* VLANs specified by the VLAN name *v_name*.

  v_range formats include number, number range, or comma-delimited list of numbers and ranges.
- ***PORT_ACTIVITY*** Ports listed in table. Options include:
  - <no parameter> table displays only active ports (same as **active-configuration** option).
  - **active-configuration** table displays only active ports.
  - **configured-ports** table displays all configured ports.

Display Values

- **VLAN** The VLAN ID.
- **Name** The name of the VLAN.
- **Status** The status of the VLAN.
- **Ports** The ports that are members of the VLAN.

Example

- This command displays status and ports of VLANs 1-1000.

```
switch>show vlan 1-1000
VLAN  Name                             Status    Ports
----- -------------------------------- --------- --------------------------------
1     default                          active    Po1
184   fet.arka                         active    Cpu, Po1, Po2
262   mgq.net                          active    PPo2, Po1
512   sant.test                        active    Cpu, Et16, Po1
821   ipv6.net                         active    Cpu, Po1, Po7

switch>
```

show vlan dynamic

The **show vlan dynamic** command displays the source and quantity of dynamic VLANs on the switch. Dynamic VLANs support VM Tracer monitoring sessions.

Command Mode

EXEC

Command Syntax

show vlan dynamic

Example

- This command displays the source and quantity of dynamic VLANs on the switch.

```
switch>show vlan dynamic
Dynamic VLAN source       VLANS
vmtracer-poc              88
switch>
```

show vlan internal allocation policy

The **show vlan internal allocation policy** command displays the method the switch uses to allocate VLANs to routed ports. The **vlan internal allocation policy** command configures the allocation method.

The allocation method consists of two configurable components:

- range: the list of VLANs that are allocated to routed ports.
- direction: the direction by which VLANs are allocated (ascending or descending).

Command Mode

EXEC

Command Syntax

```
show vlan internal allocation policy
```

Example

- This command displays the internal allocation policy.

```
switch>show vlan internal allocation policy
Internal VLAN Allocation Policy: ascending
Internal VLAN Allocation Range: 1006-4094
switch>
```

show vlan internal usage

The **show vlan internal usage** command displays the VLANs that are allocated as internal VLANs for routed ports.

A routed port is an Ethernet or port channel interface that is configured as a layer 3 interface. Routed ports do not bridge frames and are not members of any VLANs. Routed ports can have IP addresses assigned to them and packets are routed directly to and from the port.

When an interface is configured as a routed port, the switch allocates an SVI with a previously unused VLAN ID. The switch prohibits the configuration of VLANs with numbers corresponding to internal VLAN interfaces allocated to a routed port. VLAN interfaces corresponding to SVIs allocated to a routed port cannot be configured by VLAN interface configuration mode commands.

Command Mode

EXEC

Command Syntax

```
show vlan internal usage
```

Example

- This command displays the VLANs that are allocated to routed ports.

```
switch>show vlan internal usage
1006  Ethernet3
1007  Ethernet4
switch>
```

show vlan private-vlan

The **show vlan private-vlan** command displays the primary VLANs and lists the private-VLANs assigned to them. The command lists the VLAN type and attached ports configuration of the private VLANs.

Command Mode

EXEC

Command Syntax

show vlan private-vlan

Restriction

Private VLANs are available on FM4000 and Trident platform switches.

Example

- This command displays the private VLANs.

```
switch>show vlan private-vlan
Primary Secondary Type        Ports
------- --------- ----------- --------------------------------
5       25        isolated
5       26        isolated
7       31        community
7       32        isolated
switch>
```

show vlan summary

The **show vlan summary** command displays the quantity of VLANs that are configured on the switch.

Command Mode
EXEC

Command Syntax
show vlan summary

Example

- This command displays the number of VLANs on the switch.

```
switch>show vlan summary
Number of existing VLANs           : 18

switch>
```

show vlan trunk group

The **show vlan trunk group** command displays the trunk group membership of the specified VLANs.

Command Mode
EXEC

Command Syntax
show vlan [***VLAN_LIST***] **trunk group**

Parameters

- ***VLAN_LIST*** VLAN list. Options include:
  - — <no parameter> all VLANs.
  - — *v_range* VLANs specified by *v_range*.
  - — **id** *v_range* VLANs specified by *v_range*.
  - — **name** *v_name* VLANs specified by the VLAN name *v_name*.

Display Values

- **VLAN** VLAN ID.
- **Trunk Group** Trunk groups associated with the specified VLAN.

Example

- This command displays the trunk group membership of all configured VLANs.

```
switch>show vlan trunk group
VLAN     Trunk Groups
----     --------------------------------------------------------------------------
5
10       first_group
12
40       second_group
100      third_group
101      middle_group
102
200

switch>
```

state

The **state** command configures the VLAN transmission state of the configuration mode VLAN.

- **Active** state: Ports forward VLAN traffic.
- **Suspend** state: Ports block VLAN traffic.

The default transmission status is ***active***.

The **no state** command restores the default VLAN transmission state to the configuration mode VLAN by removing the corresponding **state** command from ***running-config***.

Command Mode

VLAN Configuration

Command Syntax

```
state OPERATION_STATE
no state
default state
```

Parameters

- *OPERATION_STATE* VLAN transmission state. Options include:
  - **active** VLAN traffic is forwarded
  - **suspend** LAN traffic is blocked.

Example

- These commands suspend VLAN traffic on VLANs 100-102.

```
switch(config)#vlan 100-102
switch(config-vlan-100-102)#state suspend
switch(config-vlan-100-102)#
```

switchport access vlan

The **switchport access vlan** command specifies the access VLAN of the configuration mode interface. Ethernet or port channel interfaces that are in access mode are members only of the access VLAN. Untagged frames that the interface receives are associated with the access VLAN. Frames tagged with the access VLAN are also associated with the access VLAN. The interface drops all other tagged frames that it receives. By default, VLAN 1 is the access VLAN of all Ethernet and port channel interfaces.

An interface's access mode is effective only when the interface is in access mode or dot1q-tunnel mode, as specified by the switchport mode command. Interfaces in dot1q-tunnel mode handle inbound traffic as untagged traffic and associates all traffic with the access VLAN. Interfaces configured to switchport trunk mode maintain and ignore existing switchport access commands.

The **no switchport access vlan** and **default switchport access vlan** commands restore VLAN 1 as the access VLAN of the configuration mode interface by removing the corresponding switchport access command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport access vlan v_num
no switchport access vlan
default switchport access vlan
```

Parameters

- *v_num* number of access VLAN. Value ranges from 1 to 4094. Default is 1.

Example

- These commands assign VLAN 100 as the access VLAN to Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#switchport access vlan 100
switch(config-if-Et5)#
```

switchport mode

The **switchport mode** command specifies the switching mode of the configuration mode interface. The switch supports three switching modes: access, trunk, and dot1q-tunnel.

- **Access switching mode**: The interface is a member of one VLAN, called the access VLAN, as specified by the **switchport access vlan** command. Untagged frames received on the interface are associated with the access VLAN. Tagged frames received on the interface are dropped unless they are tagged with the access VLAN. Frames transmitted from the interface are always untagged.
- **Trunk switching mode**: The interface may be a member of multiple VLANs, as configured with the **switchport trunk allowed vlan** command. Untagged traffic is associated with the interface's native VLAN, as configured with the switchport trunk native vlan command.
- **Dot1q-tunnel switching mode**: The interface treats all inbound packets as untagged traffic and handles them as traffic of its access VLAN, as specified by the **switchport access vlan** command.

The **no switchport mode** and default **default switchport mode** commands return the configuration mode interface to its default setting as an access port by deleting the corresponding **switchport mode** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport mode MODE_TYPE
no switchport mode
default switchport mode
```

Parameters

- *MODE_TYPE* switching mode of the configuration mode interfaces. Options include:
  - **access** access switching mode.
  - **dot1q-tunnel** dot1q-tunnel switching mode.
  - **trunk** trunk switching mode.

Restriction

Dot1q-tunnel switching mode is not available on Petra platform switches.

Example

- These commands configure Ethernet 4 interface as a trunk port.

```
switch(config)#interface ethernet 4
switch(config-if-Et4)#trunk
switch(config-if-Et4)#
```

switchport private-vlan mapping

The **switchport private-vlan mapping** command maps traffic received by the configuration mode interface for a specified primary VLAN to a list of secondary VLANs. Command options are available to establish a VLAN list or modify an existing list. By default, traffic to the primary VLAN is mapped to all of its secondary VLANs.

The **no switchport private-vlan mapping** and **default switchport private-vlan mapping** commands restore the default VLAN mapping by removing the corresponding **switchport private-vlan mapping** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport private-vlan mapping EDIT_ACTION
no switchport private-vlan mapping
default switchport private-vlan mapping
```

Parameters

- *EDIT_ACTION* modifications to the VLAN list.
  - — *v_range* Creates VLAN list from *v_range*.
  - — **add** *v_range* Adds specified VLANs to current list.
  - — **remove** *v_range* VLAN list contains all VLANs except those specified.

  Valid *v_range* formats include number, range, or comma-delimited list of numbers and ranges.

Restriction

Private VLANs are available on FM4000 and Trident platform switches.

Example

- These commands map Ethernet port 15 from the primary VLANs configured on the port to VLANs 5-10.

```
switch(config)#interface ethernet 15
switch(config-if-Et15)#switchport private-vlan mapping 5-10
switch(config-if-Et15)#
```

switchport trunk allowed vlan

The **switchport trunk allowed vlan** command creates or modifies the list of VLANs for which the configuration mode interface, in trunk mode, handles tagged traffic. By default, interfaces handle tagged traffic for all VLANs.

The **no switchport trunk allowed vlan** and **default switchport trunk allowed vlan** commands restores the default allowed VLAN setting of ***all*** by removing the corresponding **switchport trunk allowed vlan** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport trunk allowed vlan EDIT_ACTION
no switchport trunk allowed vlan
default switchport trunk allowed vlan
```

Parameters

- ***EDIT_ACTION*** modifications to the VLAN list.
  - — *v_range* Creates VLAN list from *v_range*.
  - — **add** *v_range* Adds specified VLANs to current list.
  - — **all** VLAN list contains all VLANs.
  - — **except** *v_range* VLAN list contains all VLANs except those specified.
  - — **none** VLAN list is empty (no VLANs).
  - — **remove** *v_range* Removes specified VLANs from current list.

  Valid *v_range* formats include number, range, or comma-delimited list of numbers and ranges.

Example

- These commands create the VLAN list of 6-10 for Ethernet interface 14, then verifies the VLAN list.

```
switch(config)#interface ethernet 14
switch(config-if-Et14)#switchport trunk allowed vlan 6-10
switch(config-if-Et14)#show interfaces ethernet 14 switchport
Name: Et14
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Access Mode VLAN: 1 (inactive)
Trunking Native Mode VLAN: 1 (inactive)
Administrative Native VLAN tagging: disabled
Trunking VLANs Enabled: 6-10
Trunk Groups:

switch(config-if-Et14)#
```

switchport trunk group

The **switchport trunk group** command assigns the configuration mode interface to the specified trunk group. Trunk group ports handle traffic of the VLANs assigned to the group.

The **no switchport switchport trunk group** and **default switchport switchport trunk group** commands remove the configuration mode interface from the specified trunk group by deleting the corresponding statement from ***running-config***. If the command does not specify a trunk group, the interface is removed from all trunk groups to which it is assigned.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport trunk group group_name
no switchport trunk group [group_name]
default switchport trunk group [group_name]
```

Parameters

- *group_name* trunk group name.

Example

- These commands assign port channel 4 to trunk group ***fe-1***.

```
switch(config)#interface port-channel 4
switch(config-if-Po4)#switchport trunk group fe-1
switch(config-if-Po4)#
```

switchport trunk native vlan

The **switchport trunk native vlan** command specifies the native VLAN for the configuration mode interface. Interfaces in trunk mode associate untagged frames with the native VLAN. Trunk mode interfaces can also be configured to drop untagged frames. The default native VLAN for all interfaces is VLAN 1.

The **no switchport trunk native vlan** and **default switchport trunk native vlan** commands restore the default native VLAN to the configuration mode interface by removing the corresponding command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport trunk native vlan VLAN_ID
no switchport trunk native vlan
default switchport trunk native vlan
```

Parameters

- ***VLAN_ID*** the ID of the native VLAN. Options include
  - v_num VLAN number. Values ranging from 1 to 4094
  - **tag** interface drops all untagged frames.

Example

- These commands configure VLAN 100 as the native VLAN for port channel 21.

```
switch(config)#interface port-channel 21
switch(config-if-Po21)#switchport trunk native vlan 100
switch(config-if-Po21)#
```

trunk group

The **trunk group** command assigns the configuration mode VLAN to a specified trunk group.

A trunk group is the set of physical interfaces that comprise the trunk and the collection of VLANs whose traffic is carried on the trunk. The traffic of a VLAN that belongs to one or more trunk groups is carried only on ports that are members of trunk groups to which the VLAN belongs. Switchport commands specify the physical interfaces that carry trunk group traffic.

The **no trunk group** and **default trunk group** commands remove the configuration mode VLAN from the specified trunk group by removing the corresponding **trunk group** statement from ***running-config***. If a trunk group is not specified, the commands remove the configuration mode VLAN from all trunk groups.

Command Mode

VLAN Configuration

Command Syntax

```
trunk group name
no trunk group [name]
default trunk group [name]
```

Parameters

- *name* a name representing the trunk group.

Example

- These commands assigns VLAN 49 to the trunk group ***mlagpeer***:

```
switch(config)#vlan 49
switch(config-vlan-49)#trunk group mlagpeer
switch(config-vlan-49)#
```

vlan

The **vlan** command places the switch in VLAN configuration mode to configure a set of virtual LANs. The command creates the specified VLANs if they do not exist prior to issuing the command. A VLAN that is in use as an internal VLAN may not be created or configured. The switch rejects any **vlan** command that specifies an internal VLAN ID.

The **default vlan** and **no vlan** commands removes the VLAN statements from ***running-config*** for the specified VLANs.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
vlan vlan_range
no vlan vlan_range
default vlan vlan_range
```

Parameters

- *vlan_range* VLAN list.

  Formats include a name, number, number range, or comma-delimited list of numbers and ranges.

Commands Available in VLAN configuration mode

- **name (VLAN configuration mode)**
- **state**
- **trunk group**

Guidelines

In MLAG configurations, VLANs operate as follows:

- The VLAN must be configured identically on both MLAG peer switches.
- The port-specific bridging configuration originates on the switch where the port is physically located. This configuration includes the switchport access VLAN, switchport mode (trunk or access), trunk-allowed VLANS, the trunk native VLAN, and the switchport trunk groups.

Example

- This command creates VLAN 49 and enters VLAN configuration mode for the new VLAN:

```
switch(config)#vlan 49
switch(config-vlan-49)#
```

vlan internal allocation policy

The **vlan internal allocation policy** command specifies the range that the switch can allocate as internal VLANs when configuring routed ports and the order of their allocation. By default, the switch allocates VLANs in ascending order from VLAN 1006 to VLAN 4094.

The **no vlan internal allocation policy** and **default vlan internal allocation policy** commands revert the policy to its default.

Command Mode

Global Configuration

Command Syntax

```
vlan internal allocation policy DIRECTION [RANGE_VLAN]
no vlan internal allocation policy
default vlan internal allocation policy
```

Parameters

- *DIRECTION* VLAN allocation number direction. Options include:
  - **ascending** allocates internal VLANs from lower VLAN bound to upper VLAN bound.
  - **descending** allocates internal VLAN from upper VLAN bound to lower VLAN bound.
- *RANGE_VLAN* allocation range. Options include:
  - <no parameter> 1006 (lower bound) to 4094 (upper bound).
  - **range** *lower upper* specifies lower bound (*lower*) and upper bound (*upper*).

Examples

- This command configures the switch to allocate internal VLANS from 3000 through 3999.

```
switch(config)#vlan internal allocation policy ascending range 3000 3999
switch(config)#
```

- This command configures the switch to allocate internal VLANS from 4094 through 1006.

```
switch(config)#vlan internal allocation policy descending
switch(config)#
```

- This command configures the switch to allocate internal VLANS from 4094 down through 4000.

```
switch(config)#vlan internal allocation policy descending range 4000 4094
switch(config)#
```

- This command reverts the allocation policy to its default (***ascending***, between ***1006*** and ***4094***).

```
switch(config)#no vlan internal allocation policy
switch(config)#
```

Chapter 12

Multi-Chassis Link Aggregation

Arista switches support Multi-Chassis Link Aggregation (MLAG) to logically aggregate ports across two switches. For example, two 10-gigabit Ethernet ports, one each from two MLAG configured switches, can connect to two 10-gigabit ports on a host, switch, or network device to create a link that appears as a single 20-gigabit port. MLAG configured ports provide Layer 2 multipathing, increased bandwidth, higher availability, and improves on traditional active-passive or Spanning Tree governed infrastructures.

The Multi-Chassis Link Aggregation chapter contains these sections:

- Section 12.1: MLAG Introduction
- Section 12.2: MLAG Conceptual Overview
- Section 12.3: MLAG Maintenance
- Section 12.4: Configuring MLAG
- Section 12.5: MLAG Implementation Example
- Section 12.6: MLAG Commands

12.1 MLAG Introduction

High availability data center topologies typically provide redundancy protection at the expense of oversubscription by connecting top-of-rack (TOR) switches and servers to dual aggregation switches. In these topologies, Spanning Tree Protocol prevents network loops by blocking half of the links to the aggregation switches. This reduces the available bandwidth by 50%.

Deploying MLAG removes oversubscription by configuring an MLAG link between two aggregation switches to create a single logical switching instance that utilizes all connections to the switches. Interfaces on both devices participate in a distributed port channel, enabling all active paths to carry data traffic while maintaining the integrity of the Spanning Tree topology.

MLAG provides these benefits:

- Provides higher bandwidth links as network traffic increases.
- Utilizes bandwidth more efficiently with fewer uplinks blocked by STP.
- Connects to other switches and servers by static LAG or LACP without other proprietary protocols.
- Aggregates up to 32 10-Gb Ethernet ports across two switches: 16 ports from each switch.
- Supports normal STP operation to prevent loops.
- Supports active-active Layer-2 redundancy.

12.2 MLAG Conceptual Overview

12.2.1 MLAG Operation Process

A multichassis link aggregation group (MLAG) is a pair of links that terminate on two cooperating switches and appear as an ordinary link aggregation group (LAG). The cooperating switches are MLAG peer switches and communicate through an interface called a peer link. While the peer link's primary purpose is exchanging MLAG control information between peer switches, it also carries data traffic from devices that are attached to only one MLAG peer and have no alternative path. An MLAG domain consists of the peer switches and the control links that connect the switches.

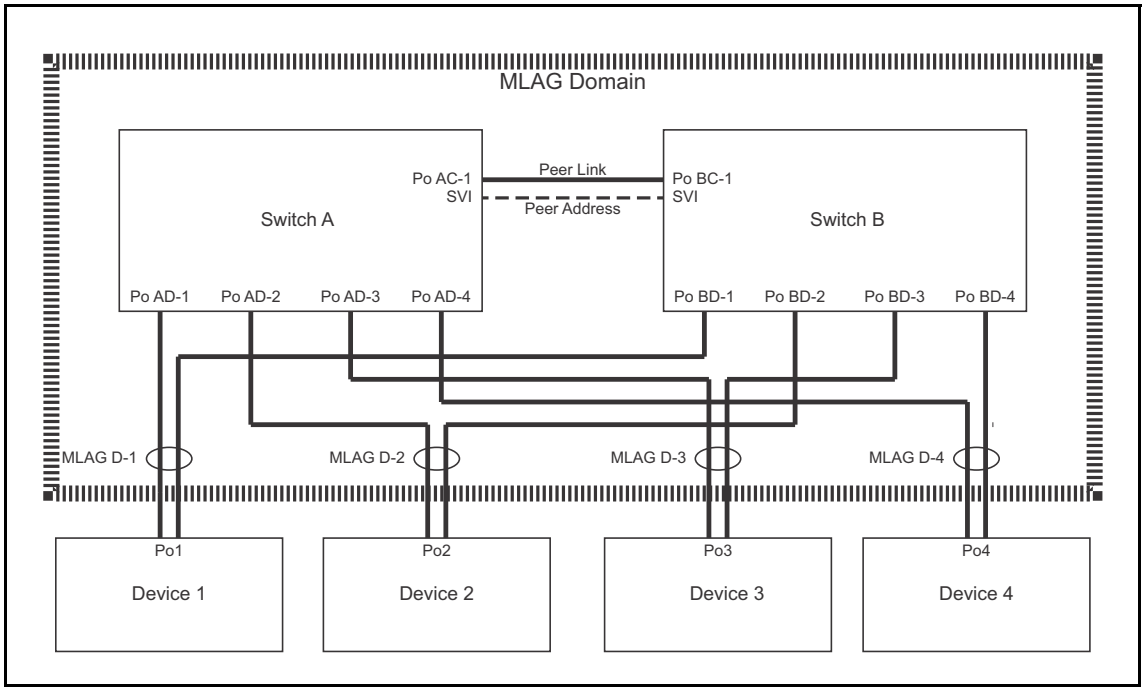
In Figure 12-1, Switch A and Switch B are peer switches in the MLAG domain and connect to each other through the peer link. Each peer switch uses the peer address to form and maintain the peer link.

The MLAG domain ID is a text string configured in each peer switch. MLAG switches use this string to identify their peers. The MLAG System ID (MSI) is the MLAG domain's MAC address. The MSI is automatically derived when the MLAG forms and does not match the bridge MAC address of either peer. Each peer uses the MSI in STP and LACP PDUs.

The topology in Figure 12-1 contains four MLAGs: one MLAG connects each device to the MLAG domain. Each peer switch connects to the four servers through MLAG link interfaces.

In a conventional topology, with dually-attaching devices to multiple switches for redundancy, Spanning Tree Protocol (STP) blocks half of the switch-device links. In the MLAG topology, STP does not block any portion because it views the MLAG Domain as a single switch and each MLAG as a single port. The MLAG protocol facilitates the balancing of device traffic between the peer switches.

Figure 12-1 MLAG Domain Topology



When MLAG is disabled, peer switches revert to their independent state. MLAG is disabled by any of the following:

- MLAG configuration changes.

- The TCP connection breaks.
- The peer-link or local-interface goes down.
- A switch does not receive a response to a keep alive message from its peer within a specified period.

12.2.2 MLAG Interoperability with Other Features

The following sections describe MLAG interaction with other switch features.

12.2.2.1 VLANs

VLAN parameters must be configured identically on each peer for the LAGs comprising the peer link and MLAGs. These parameters include the switchport access VLAN, switchport mode, trunk-allowed VLANs, the trunk native VLAN, and switchport trunk groups. Configuration discrepancies may result in traffic loss in certain failure scenarios. Port-specific bridging configuration originates on the switch where the port is physically located.

12.2.2.2 LACP

Link Aggregation Control Protocol (LACP) should be used on all MLAG interfaces, including the peer-link. LACP control packets reference the MLAG system ID.

12.2.2.3 Static MAC Addresses

A static MAC address configured on an MLAG interface is automatically configured on the peer's corresponding interface. Configuring static MAC addresses on both peers prevents undesired flooding if an MLAG peer relationship fails.

If the MLAG peer relationship is broken or if all local members of an MLAG port channel go down, the peer is no longer automatically configured with the static MAC address.

Static MAC addresses configured as drop MAC entries are not shared between peers when unicast MAC address filtering on the switch is enabled to drop traffic with a specific source or destination MAC address.

12.2.2.4 STP

When implementing MLAG in a spanning tree network, spanning tree must be configured globally and on port-channels configured with an MLAG ID. Port specific spanning tree configuration comes from the switch where the port physically resides. This includes spanning-tree PortFast BPDU Guard and BPDU filter.

12.3 MLAG Maintenance

These sections describe tasks required for MLAG to operate on the switch:

- Section 12.3.1: Verifying Control Plane ACL Compatibility
- Section 12.3.2: MLAG Availability through a Single Functional Peer
- Section 12.3.3: Upgrading MLAG Peers

12.3.1 Verifying Control Plane ACL Compatibility

The control plane access control list (ACL) must be configured to allow only the peer link neighbor to generate MLAG control traffic. The required rules are included in the default ACL for the control plane.

These two rules are required in the control plane ACL:

```
permit tcp any any eq mlag ttl eq 255
permit udp any any eq mlag ttl eq 255
```

To verify these rules are in the control plane ACL, issue the **show ip access-lists** command. In the following example, the required rules are in lines 60 and 70:

```
Switch#show ip access-lists
IP Access List default-control-plane-acl [readonly]
     10 permit icmp any any [match 10, 1 day, 2:50:33 ago]
     20 permit ip any any tracked [match 3501, 7 days, 0:21:39 ago]
     30 permit ospf any any
     40 permit tcp any any eq ssh telnet www snmp bgp https [match 12, 1 day,
2:20:22 ago]
     50 permit udp any any eq bootps bootpc snmp [match 242, 7 days, 2:41:14 ago]
     60 permit tcp any any eq mlag ttl eq 255
     70 permit udp any any eq mlag ttl eq 255
     80 permit vrrp any any
     90 permit ahp any any
```

MLAG peers that function as routers must each have routing enabled.

12.3.2 MLAG Availability through a Single Functional Peer

MLAG high availability advantages are fully realized when all devices that connect to one MLAG switch also connect to the peer switch. A switch can continue supporting MLAG when its peer is offline if the STP agent is restartable. When one peer is offline, data traffic flows from the devices through the MLAG component link that connects to the functioning switch. When a switch is offline, its interfaces and ports do not appear in **show mlag** and ***show spanning tree protocol*** commands of the functioning peer.

To view the restartability status of the STP agent, issue the **show spanning-tree bridge detail** command:

```
switch-1#show spanning-tree bridge detail | grep agent
Stp agent is restartable
```

STP agent restartability requires consistent configuration between the peers of STP, LACP, MLAG, and switchport parameters. Events triggering an STP state machine change may also briefly prevent the STP agent from being restartable.

If an MLAG peer reboots, all ports except those in the peer-link port-channel remain in ***errdisabled*** state for a specified period. This period allows all topology states to stabilize before the switch begins forwarding traffic. The specified period is configured by the **reload-delay** command. The default period is 5 minutes; the recommended minimum value required to ensure the forwarding hardware is initialized with the topology state depends on the switch platform:

- fixed configuration switches: 300 seconds
- modular switches: 600 seconds

Severing the physical connection (cable) that establishes the peer-link between MLAG peers may result in a ***split brain*** state where each peer independently enters spanning tree state to prevent topology loops. Sessions established through one interface of a dual attached device may fail if its path is disrupted by the STP reconvergence, possibly resulting in temporarily lost connectivity. Sessions can be reestablished if permitted by the resulting topology.

12.3.3 Upgrading MLAG Peers

MLAG ISSU (In-Service Software Upgrade) upgrades EOS software on one MLAG peer with minimal traffic disruptions on active MLAG interfaces and without changing the network topology.

12.3.3.1 Verifying Configuration Compatibility

A seamless EOS upgrade on an MLAG peer requires that the following features are configured consistently on each switch:

- VLANs
- Switchport configution on port channel interfaces that are configured with an MLAG ID
- STP configuration (global)

12.3.3.2 Version Compatibility

A switch running MLAG can be upgraded without disrupting MLAG traffic when the upgrade EOS version is compatible with the version on the peer switch. Table 12-1 displays compatible EOS versions.

EOS Version	Previously Released Compatible EOS Versions
4.10.0	4.9.x, where x>=2
4.10.1	4.9.x, where x>=2, 4.10.0
4.10.2	4.9.x, where x>=2, 4.10.0, 4.10.1

Table 12-1 EOS Version Compatibility Matrix

12.3.3.3 Reload Warning Conditions

Entering an EOS reload command while MLAG is active generates warning messages if conditions that can result in packet loss during the upgrade are present. All warnings should be resolved before confirming the reload request. Table 12-2 displays the reload conditions and a common resolution method for each condition..

Reload Condition	Resolution Method
Compatibility check	Refer to Table 12-1 to verify that the new version is compatible with the currently installed version
Active-partial MLAG warning	Bring up the the remote port-channel. If the MLAG is not actively used, then this warning can be ignored.
STP is not restartable	Wait for STP to be restartable: typically 30 seconds, up to 120 seconds for a newly started TP agent. Refer to Section 12.3.2 for information on checking restartability.
Reload delay too low	Configure a reload delay value greater than or equal to the default. Recommend delay is 300 seconds for TOR switches and 600 seconds for modulars.
Peer has errdisabled interfaces	Wait for reload-delay to expire on the peer.

Table 12-2 Reload Warning Resolutions

Example

- The following **reload** command generates MLAG warning conditions that should be addressed before confirming the ***proceed with reload*** prompt.

```
switch(config)#reload
If you are performing an upgrade, and the Release Notes for the
new version of EOS indicate that MLAG is not backwards-compatible
with the currently installed version (4.9.2), the upgrade will
result in packet loss.

The following MLAGs are not in Active mode. Traffic to or from these ports will
be lost during the upgrade process.
                                                                    local/remote
  mlag      desc               state      local      remote           status
--------- --------- ------------------- ---------- ----------- ------------
    14                active-partial       Po14        Po14         up/down
    15                active-partial       Po15        Po15         up/down

Stp is not restartable. Topology changes will occur during the upgrade process.

The configured reload delay of 100 seconds is below the default
value of 300 seconds. A longer reload delay allows more time to
rollback an unsuccessful upgrade due to incompatibility.

The other MLAG peer has errdisabled interfaces. Traffic loss will occur during
the upgrade process.

Proceed with reload? [confirm]
```

12.3.3.4 Performing an MLAG ISSU Upgrade

The following procedure performs an MLAG ISSU upgrade:

Step 1 Verfy configuration consistency on each peer (Section 12.3.3.1).

Step 2 Verify version compatibility between the new and existing images (Section 12.3.3.2).

Step 3 Configure **reload-delay** (Section 12.3.2). Delay recommendation varies by switch type.

Step 4 Install the new image onto one of the peers:

Step a Upload the new image to the switch (Section 2.4.1).

Step b Set the boot path to the new image (Section 2.4.2).

Step c Enter the **reload** command. (Section 2.4.3).

Step 5 Resolve all reload warnings (Section 12.3.3.3).

Step 6 Confirm the reload.

Step 7 Wait for MLAG peers to renegotiate to the active state and reload-delay expiry on rebooted peer.

Step 8 Repeat the upgrade process for the other peer.

When upgrading modular switches with dual supervisors, upgrade the standby supervisors first, then upgrade the active supervisors.

12.4 Configuring MLAG

These sections describe the basic MLAG configuration steps:

- Section 12.4.1: Configuring the MLAG Peers
- Section 12.4.2: Configuring MLAG Services

12.4.1 Configuring the MLAG Peers

Connecting two switches as MLAG peers requires the establishment of the peer link and an SVI that defines local and peer IP addresses on each switch.

The peer link is composed of a LAG between the switches. When all devices that connect to the MLAG domain are dually connected to the switches through an MLAG, a peer link of two Ethernet interfaces is sufficient to handle MLAG control data and provide N+1 redundancy. When the domain connects to devices through only one MLAG peer, the peer link may require additional Ethernet interfaces to manage data traffic.

The steps that configure two switches as MLAG peers include:

- Configuring the Port Channels, VLAN Interfaces, and IP addresses
- Configure Peer Parameters

12.4.1.1 Configuring the Port Channels, VLAN Interfaces, and IP addresses

The peer link is a normal port channel. The local address is the SVI that maps to the peer link port channel. The port channel and SVI must be configured on each peer switch. The port channel should be an active LACP port. The local and peer addresses must be located on the same IP address subnet. Autostate should be disabled on the SVI configured as the local interface.

Examples

- These commands create a port channel interface from two Ethernet interfaces and configure it as a trunk group on each switch. Each port channel is configured as an active LACP port.

  Switch 1

```
Switch1#config
Switch1(config)#interface ethernet 1-2
Switch1(config-if-Et1-2)#channel-group 10 mode active
Switch1(config-if-Et1-2)#interface port-channel 10
Switch1(config-if-Po10)#switchport mode trunk
Switch1(config-if-Po10)#switchport trunk group m1peer
Switch1(config-if-Po10)#exit
Switch1(config)#
```

  Switch 2

```
Switch2#config
Switch2(config)#interface ethernet 1-2
Switch2(config-if-Et1-2)#channel-group 10 mode active
Switch2(config-if-Et1-2)#interface port-channel 10
Switch2(config-if-Po10)#switchport mode trunk
Switch2(config-if-Po10)#switchport trunk group m2peer
Switch2(config-if-Po10)#exit
Switch2(config)#
```

- These commands create an SVI for the local interface and associate it to the trunk group assigned to the peer link port channel. STP is disabled for the peer link VLAN. The SVI creates a Layer 3 endpoint in the switch and enables MLAG processes to communicate with TCP. The IP address can be any unicast address that does not conflict with other SVIs.

  Switch 1

```
Switch1#config
Switch1(config)#vlan 4094
Switch1(config-vlan-4094)#trunk group m1peer
Switch1(config-vlan-4094)#interface vlan 4094
Switch1(config-if-Vl4094)#ip address 10.0.0.1/30
Switch1(config-if-Vl4094)#no autostate
Switch1(config-if-Vl4094)#exit
Switch1(config)#no spanning-tree vlan 4094
Switch1(config)#
```

  Switch 2

```
Switch2#config
Switch2(config)#vlan 4094
Switch2(config-vlan-4094)#trunk group m2peer
Switch2(config-vlan-4094)#interface vlan 4094
Switch2(config-if-Vl4094)#ip address 10.0.0.2/30
Switch2(config-if-Vl4094)#no autostate
Switch2(config-if-Vl4094)#exit
Switch2(config)#no spanning-tree vlan 4094
Switch2(config)#
```

12.4.1.2 Configure Peer Parameters

Peer connection parameters configure the connection between the MLAG peer switches. This section describes the following peer configuration parameters.

- MLAG Configuration Mode
- Local VLAN Interface
- Peer Address
- Peer Link
- Domain ID
- Heartbeat Interval and Timeout
- Reload Delay Period

MLAG Configuration Mode

Peer connection parameters are configured in mlag-configuration mode. The **mlag configuration (global configuration)** command places the switch in MLAG configuration mode.

Example

- This command places the switch in MLAG configuration mode.

```
Switch(config)#mlag configuration
Switch(config-mlag)#
```

Local VLAN Interface

The local interface specifies the SVI upon which the switch sends MLAG control traffic. The local IP address is specified within the definition of the VLAN associated with the local interface. The Peer Address configures the control traffic destination on the peer switch.

The **local-interface** command specifies a VLAN interface as the peer link SVI.

Example

- This command configures VLAN 4094 as the local interface.

```
Switch(config-mlag)#local-interface vlan 4094
Switch(config-mlag)#
```

Peer Address

The peer address is the destination address on the peer switch for MLAG control traffic. If the peer IP address is unreachable, MLAG peering fails and both peer switches revert to their independent state.

The **peer-address** command specifies the peer address.

Example

- This command configures a peer address of 10.0.0.2.

```
Switch(config-mlag)#peer-address 10.0.0.2
Switch(config-mlag)#
```

Peer Link

An MLAG is formed by connecting two switches through an interface called a peer link. The peer link carries MLAG advertisements, keepalive messages, and data traffic between the switches. This information keeps the two switches working together as one. While interfaces comprising the peer links on each switch must be compatible, they need not use the same interface number. Ethernet and Port-channel interfaces can be configured as peer links.

The **peer-link** command specifies the interface through which the switch communicates MLAG control traffic.

Example

- This command configures port-channel 10 as the peer link.

```
Switch(config-mlag)#peer-link port-channel 10
Switch(config-mlag)#
```

Domain ID

The MLAG domain ID is a unique identifier for an MLAG domain. The MLAG domain ID must be the identical on each switch to facilitate MLAG communication.

The **domain-id** command configures the MLAG domain ID.

Example

- This command configures ***mlagDomain*** as the domain ID:

```
Switch(config-mlag)#domain-id mlagDomain
Switch(config-mlag)#
```

Heartbeat Interval and Timeout

The heartbeat interval specifies the period between the transmission of successive keepalive messages. Each MLAG switch transmits keepalive messages and monitors message reception from its peer. The heartbeat timeout is reset when the switch receives a keepalive message. If the heartbeat timeout expires, the switch disables MLAG under the premise that the peer switch is not functioning.

The **heartbeat-interval** command configures the heartbeat interval between 1 and 30 seconds, with a default value of 2 seconds. The heartbeat timeout expiry is 2.5 times the heartbeat interval.

Example

- This command configures the heartbeat interval as 2.5 seconds (2500 ms).

```
Switch(config-mlag)#heartbeat-interval 2500
Switch(config-mlag)#
```

Reload Delay Period

The reload delay period specifies the interval that non-peer links are disabled after an MLAG peer reboots. This interval allows non-peer links to learn multicast and OSPF states before the ports start handling traffic. The recommended minimum value required to ensure the forwarding hardware is initialized with the topology state depends on the switch platform:

- fixed configuration switches: 300 seconds
- modular switches: 600 seconds

The **reload-delay** command configures the reload delay period. The reload delay period varies between 0 seconds and one hour (3600 seconds) with a default period is five minutes

Example

- This command configures the reload delay interval as 2.5 minutes (150 seconds).

```
Switch(config-mlag)#reload-delay 150
Switch(config-mlag)#
```

Shutdown

The **shutdown (MLAG)** command (MLAG configuration mode) disables MLAG operations without disrupting the MLAG configuration. The **no mlag configuration** command (global configuration mode) disables MLAG and removes the MLAG configuration.

The **no shutdown** command resumes MLAG activity.

Examples

- This command disables MLAG activity on the switch.

```
Switch(config-mlag)#shutdown
Switch(config-mlag)#
```

- This command resumes MLAG activity on the switch.

```
Switch(config-mlag)#no shutdown
Switch(config-mlag)#
```

12.4.2 Configuring MLAG Services

An MLAG is a pair of links that originate on a network attached device and terminate on the two MLAG peer switches. The MLAG switches coordinate traffic to the device through a common **mlag (port-channel interface configuration)** command on the interfaces that connect to the device.

The MLAG ID differs from the MLAG domain ID. The MLAG domain ID is assigned globally per switch in MLAG Configuration mode, and the same MLAG domain ID must be on both switches.

It is not recommended to use MLAGs in conjunction with static LAGs. Configure the downstream switch or router connected to the MLAG peers to negotiate a LAG with LACP. For Arista Networks switches, this is in respect to a configuration such as **channel-group** *group-number* **mode on**.

Port channels that are configured as an MLAG must have identical port channel numbers. Although the MLAG ID is a distinct parameter from the port channel number, best practices recommend assigning the MLAG ID to match the port channel number.

The following example does not follow this convention to emphasize the parameters that are distinct. The example in Section 12.5 follows the best practices convention.

Example

- These Switch 1 commands bundle Ethernet interfaces 3 and 4 in port channel 20, then associates that port-channel with MLAG 12.

```
Switch1(config)#interface ethernet 3-4
Switch1(config-if-Et3-4)#channel-group 20 mode active
Switch1(config-if-Et3-4)#interface port-channel 20
Switch1(config-if-Po20)#mlag 12
Switch1(config-if-Po20)#exit
Switch1(config)#
```

- These Switch-2 commands bundle Ethernet interfaces 9 and 10 in port channel 20, then associates that port-channel with MLAG 12.

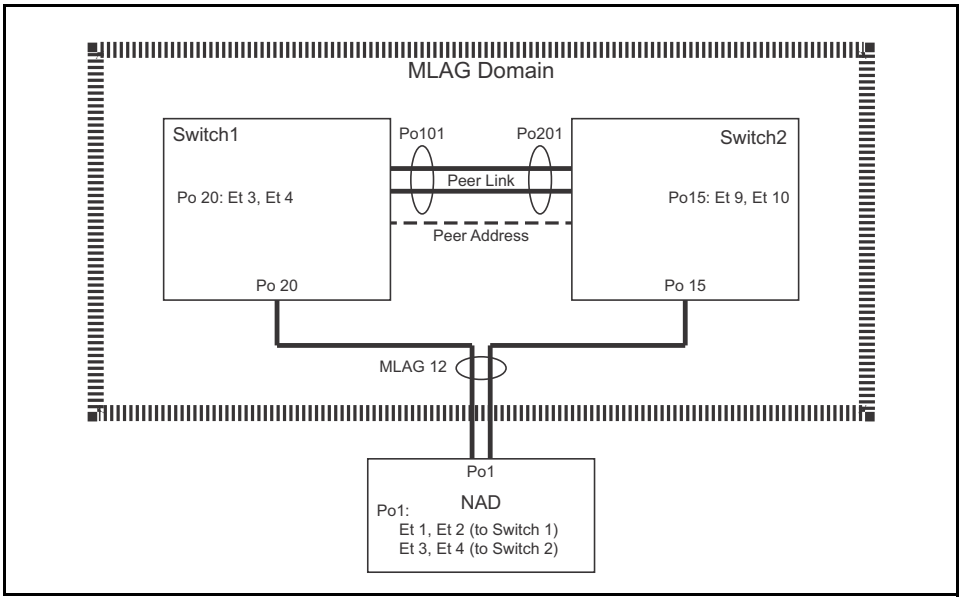
```
Switch2(config)#interface ethernet 9-10
Switch2(config-if-Et3-4)#channel-group 20 mode active
Switch2(config-if-Et3-4)#interface port-channel 15
Switch2(config-if-Po15)#mlag 12
Switch2(config-if-Po15)#exit
Switch2(config)#
```

- These commands configure the port channels that attach to the MLAG on network attached device:

```
NAD(config)#interface ethernet 1-4
NAD(config-if-Et1-4)#channel-group 1 mode active
NAD(config-if-Et1-4)#exit
NAD(config)#
```

Figure 12-2 displays the result of the interface MLAG configuration.

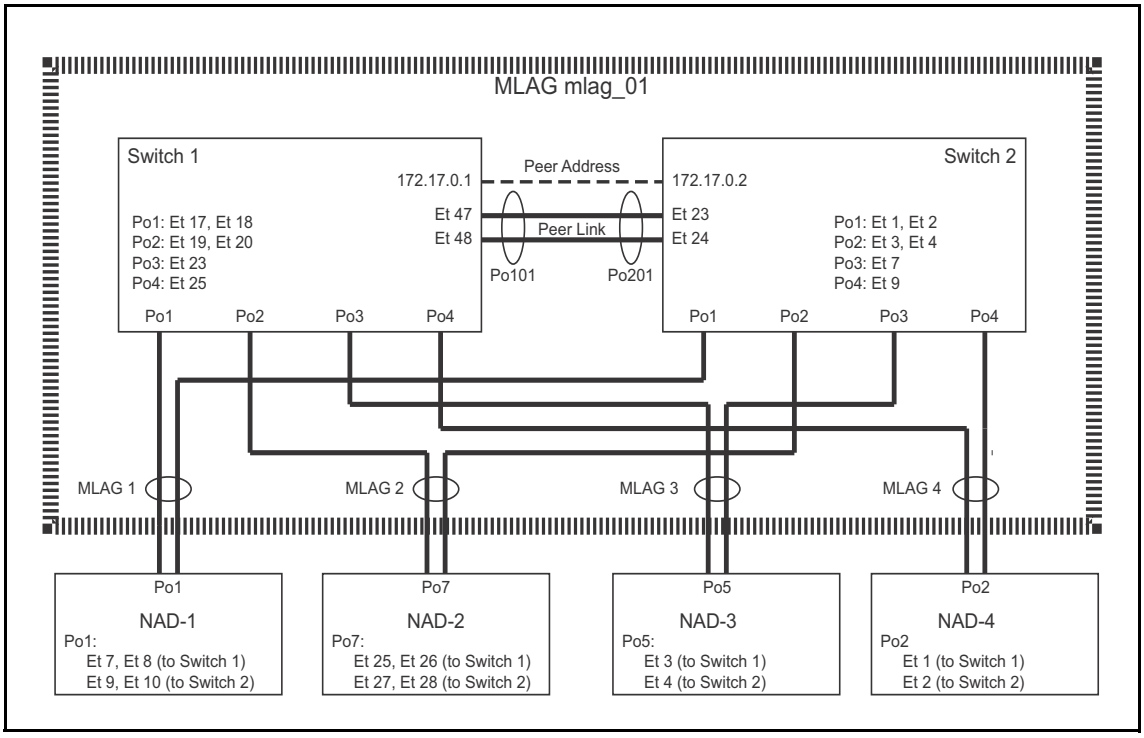
Figure 12-2 MLAG Interface Configuration



12.5 MLAG Implementation Example

This example creates an MLAG Domain, then configures MLAG connections between the peer switches and four Network Attached Devices (NADs). The MLAG switches connect through a LAG and communicate with the NADs through MLAGs. Although the NADs can be any device that supports LACP LAGs, the devices in this example are Arista switches.

Figure 12-3 MLAG Implementation Example



12.5.1 Topology

Figure 12-3 displays the MLAG topology. Switch 1 and Switch 2 are MLAG peers that logically represent a single Layer 2 switch. The peer link between the switches contains the following interfaces:

- Switch 1: Ethernet 47, Ethernet 48
- Switch 2: Ethernet 23, Ethernet 24

The example configures MLAGs from the MLAG Domain to four network attached devices (NAD-1, NAD-2, NAD-3, NAD-4).

12.5.2 Configuring the Peer Switch Connections

To configure the switches in the described topology, perform the tasks in these sections:

- Section 12.5.2.1: Configuring the Peer Switch Port Channels
- Section 12.5.2.2: Configuring the Peer Switch SVIs
- Section 12.5.2.3: Configuring the Peer Links

12.5.2.1 Configuring the Peer Switch Port Channels

These commands create the port channels the switches use to establish the peer link.

These commands create port channels on Switch1

```
Switch1(config)#interface ethernet 47-48
Switch1(config-if-Et47-48)#channel-group 101 mode active
Switch1(config-if-Et47-48)#interface port-channel 101
Switch1(config-if-Po101)#switchport mode trunk
Switch1(config-if-Po101)#switchport trunk group peertrunk
Switch1(config-if-Po101)#exit
Switch1(config)#
```

These commands create port channels on Switch2

```
Switch2(config)#interface ethernet 23-24
Switch2(config-if-Et23-24)#channel-group 201 mode active
Switch2(config-if-Et23-24)#interface port-channel 201
Switch2(config-if-Po201)#switchport mode trunk
Switch2(config-if-Po201)#switchport trunk group trunkpeer
Switch2(config-if-Po201)#exit
Switch2(config)#
```

12.5.2.2 Configuring the Peer Switch SVIs

For each peer switch, these commands create an SVI and associate it to the trunk group assigned to the peer link port channel. STP is disabled on the VLAN.

These commands configure the SVI on Switch1

```
Switch1(config)#vlan 4094
Switch1(config-vlan-4094)#trunk group peertrunk
Switch1(config-vlan-4094)#interface vlan 4094
Switch1(config-if-Vl4094)#ip address 172.17.0.1/30
Switch1(config-if-Vl4094)#no autostate
Switch1(config-if-Vl4094)#exit
Switch1(config)#no spanning-tree vlan 4094
Switch1(config)#
```

These commands configure the SVI on Switch2

```
Switch2(config)#vlan 4094
Switch2(config-vlan-4094)#trunk group trunkpeer
Switch2(config-vlan-4094)#interface vlan 4094
Switch2(config-if-Vl4094)#ip address 172.17.0.2/30
Switch2(config-if-Vl4094)#no autostate
Switch2(config-if-Vl4094)#exit
Switch2(config)#no spanning-tree vlan 4094
Switch2(config)#
```

12.5.2.3 Configuring the Peer Links

These commands create the peer links on each MLAG switch.

These commands create peer links on Switch1

```
Switch1(config)#mlag configuration
Switch1(config-mlag)#local-interface vlan 4094
Switch1(config-mlag)#peer-address 172.17.0.2
Switch1(config-mlag)#peer-link port-channel 101
Switch1(config-mlag)#domain-id mlag_01
Switch1(config-mlag)#heartbeat-interval 2500
Switch1(config-mlag)#reload-delay 150
Switch1(config-mlag)#exit
Switch2(config)#
```

These commands create peer links on Switch2

```
Switch2(config)#mlag configuration
Switch2(config-mlag)#local-interface vlan 4094
Switch2(config-mlag)#peer-address 172.17.0.1
Switch2(config-mlag)#peer-link port-channel 201
Switch2(config-mlag)#domain-id mlag_01
Switch2(config-mlag)#heartbeat-interval 2500
Switch2(config-mlag)#reload-delay 150
Switch2(config-mlag)#exit
Switch2(config)#
```

12.5.3 Configuring Peer Switch MLAGs

These commands create the MLAGs that connect the MLAG domain to the network attached devices.

These commands configure MLAG 1 on Switch1

```
Switch1(config)#interface ethernet 17-18
Switch1(config-if-Et17-18)#channel-group 1 mode active
Switch1(config-if-Et17-18)#interface port-channel 1
Switch1(config-if-Po1)#mlag 1
Switch1(config-if-Po1)#exit
Switch1(config)#
```

These commands configure MLAG 1 on Switch2

```
Switch2(config)#interface ethernet 1-2
Switch2(config-if-Et1-2)#channel-group 1 mode active
Switch2(config-if-Et1-2)#interface port-channel 1
Switch2(config-if-Po1)#mlag 1
Switch2(config-if-Po1)#exit
Switch2(config)#
```

These commands configure MLAG 2 on Switch1

```
Switch1(config)#interface ethernet 19-20
Switch1(config-if-Et19-20)#channel-group 2 mode active
Switch1(config-if-Et19-20)#interface port-channel 2
Switch1(config-if-Po2)#mlag 2
Switch1(config-if-Po2)#exit
Switch1(config)#
```

These commands configure MLAG 2 on Switch2

```
Switch2(config)#interface ethernet 3-4
Switch2(config-if-Et3-4)#channel-group 2 mode active
Switch2(config-if-Et3-4)#interface port-channel 2
Switch2(config-if-Po2)#mlag 2
Switch2(config-if-Po2)#exit
Switch2(config)#
```

These commands configure MLAG 3 on Switch1

```
Switch1(config)#interface ethernet 23
Switch1(config-if-Et23)#channel-group 3 mode active
Switch1(config-if-Et23)#interface port-channel 3
Switch1(config-if-Po3)#mlag 3
Switch1(config-if-Po3)#exit
Switch1(config)#
```

These commands configure MLAG 3 on Switch2

```
Switch2(config)#interface ethernet 7
Switch2(config-if-Et7)#channel-group 3 mode active
Switch2(config-if-Et7)#interface port-channel 3
Switch2(config-if-Po3)#mlag 3
Switch2(config-if-Po3)#exit
Switch2(config)#
```

These commands configure MLAG 4 on Switch1

```
Switch1(config)#interface ethernet 25
Switch1(config-if-Et25)#channel-group 4 mode active
Switch1(config-if-Et25)#interface port-channel 4
Switch1(config-if-Po4)#mlag 4
Switch1(config-if-Po4)#exit
Switch1(config)#
```

These commands configure MLAG 4 on Switch2

```
Switch2(config)#interface ethernet 9
Switch2(config-if-Et9)#channel-group 4 mode active
Switch2(config-if-Et9)#interface port-channel 4
Switch2(config-if-Po4)#mlag 4
Switch2(config-if-Po4)#exit
Switch2(config)#
```

12.5.4 Configuring the Network Attached Devices

These commands create the LAGs on the Network Attached Devices that connect to the MLAG domain.

These commands configure the port channels on NAD-1

```
NAD-1(config)#interface ethernet 7-10
NAD-1(config-if-Et7-10)#channel-group 1 mode active
NAD-1(config-if-Et7-10)#exit
NAD-1(config)#
```

These commands configure the port channels on NAD-2

```
NAD-2(config)#interface ethernet 25-28
NAD-2(config-if-Et25-28)#channel-group 7 mode active
NAD-2(config-if-Et25-28)#exit
NAD-2(config)#
```

These commands configure the port channels on NAD-3

```
NAD-3(config)#interface ethernet 3-4
NAD-3(config-if-Et3-4)#channel-group 5 mode active
NAD-3(config-if-Et3-4)#exit
NAD-3(config)#
```

These commands configure the port channels on NAD-4

```
NAD-4(config)#interface ethernet 1-2
NAD-4(config-if-Et1-2)#channel-group 2 mode active
NAD-4(config-if-Et1-2)#exit
NAD-4(config)#
```

12.5.5 Verification

The following tasks verify the MLAG peer and connection configuration:

- Section 12.5.5.1: Verify the Peer Switch Connection
- Section 12.5.5.2: Verify the MLAGs
- Section 12.5.5.3: Verify Spanning Tree Protocol (STP)
- Section 12.5.5.4: Verify the MLAG Port Channel
- Section 12.5.5.5: Verify the VLAN Membership

12.5.5.1 Verify the Peer Switch Connection

To display the MLAG configuration and the MLAG status on Switch 1, use the **show mlag** command:

```
Switch1#show mlag
MLAG Configuration:
domain-id          :            mlag_01
local-interface    :           Vlan4094
peer-address       :        172.17.0.2
peer-link          :    Port-Channel101

MLAG Status:
state              :             Active
peer-link status   :                 Up
local-int status   :                 Up
system-id          :  02:1c:FF:00:15:38

MLAG Ports:
Disabled           :   0
Configured         :   0
Inactive           :   0
Active-partial     :   0
Active-full        :   4
```

To display the MLAG configuration and the MLAG status on Switch 2, use the **show mlag** command:

```
Switch2#show mlag
MLAG Configuration:
domain-id          :            mlag_01
local-interface    :           Vlan4094
peer-address       :        172.17.0.1
peer-link          :    Port-Channel102

MLAG Status:
state              :             Active
peer-link status   :                 Up
local-int status   :                 Up
system-id          :  02:1c:FF:00:15:41

MLAG Ports:
Disabled           :   0
Configured         :   0
Inactive           :   0
Active-partial     :   0
Active-full        :   4
```

12.5.5.2 Verify the MLAGs

The **show mlag interfaces** command displays MLAG connections between the MLAG switches and the Network Attached Devices

- This **show mlag interfaces** command displays MLAG connections between the MLAG peer Switch 1 and the network attached devices:

```
Switch1#show mlag interfaces
                                                                  local/remote
  mlag     desc                    state     local     remote         status
-------------------------------------------------------------------------------
     1     sw1.po1          active-full       Po1       Po1          up/up
     2     sw1.po2          active-full       Po2       Po2          up/up
     3     sw1.po3          active-full       Po3       Po3          up/up
     4     sw1.po4          active-full       Po4       Po4          up/up
```

- The following **show mlag interfaces** command, with the ***detail*** option, displays MLAG connections between the MLAG peer Switch 1 and the network attached devices

```
Switch2#show mlag interfaces detail
                                local/remote
mlag       state local remote   oper   config          last change  changes
-------------------------------------------------------------------------------
   1 active-full   Po1    Po1  up/up  ena/ena  6 days, 2:08:28 ago        5
   2 active-full   Po2    Po2  up/up  ena/ena  6 days, 2:08:30 ago        5
   3 active-full   Po3    Po3  up/up  ena/ena  6 days, 2:08:33 ago        5
   4 active-full   Po4    Po4  up/up  ena/ena  6 days, 2:08:41 ago        5
Switch2#
```

12.5.5.3 Verify Spanning Tree Protocol (STP)

STP functions and can be displayed from each peer switch. MLAG interfaces are displayed as a single entry. Configured interfaces on each switch that are not included in an MLAG are displayed. Local interfaces have the normal notation; remote interfaces are preceded by ***P*** or ***Peer***.

VLAN Output 1: Assume VLAN 3903 includes MLAG 1

```
Switch1#show spanning-tree vlan 3903
Spanning tree instance for vlan 3903
VL3903
  Spanning tree enabled protocol rapid-pvst
  Root ID    Priority    36671
             Address     001c.730c.3009
             Cost        1999 (Ext) 0 (Int)
             Port        105 (Port-Channel5)
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    36671  (priority 32768 sys-id-ext 3903)
             Address     021c.7300.1319
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Po1              root       forwarding 1999      128.105  P2p
Switch1#
```

The output displays MLAG 1 under its local interface name (Po1). A peer interface is not displayed because spanning tree considers the local and remote Port Channels as a single MLAG interface.

VLAN Output 2: Assume VLAN 3908 does not include any MLAGs

```
Switch1#show spanning-tree vlan 3908
Spanning tree instance for vlan 3908
VL3908
  Spanning tree enabled protocol rapid-pvst
  Root ID    Priority    36676
             Address     021c.7300.1319
             This bridge is the root

  Bridge ID  Priority    36676  (priority 32768 sys-id-ext 3908)
             Address     021c.7300.1319
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et17             designated forwarding 2000      128.217  P2p
Et18             designated forwarding 2000      128.218  P2p
PEt17            designated forwarding 2000      128.17   P2p
PEt18            designated forwarding 2000      128.18   P2p
```

The output displays all interfaces from both switches. Each interface is explicitly displayed because they are individual units that STP must consider when selecting ports to block.

- Et17 and Et18 are located on the switch where the **show spanning-tree** command is issued.
- PEt17 and PEt18 are located on the remote switch from where the command was issued

An identical command issued on the peer switch displays similar information.

Verify the MLAG does not create topology loops (show spanning-tree blocked)

```
Switch1#show spanning-tree blocked
Name       Blocked Interfaces List
---------- --------------------------------------------------------------------

Number of blocked ports (segments) in the system : 0
Switch1#
```

12.5.5.4 Verify the MLAG Port Channel

Issue the command **show port-channel** for channel 1-4 from Switch 1:

```
Switch#show port-channel 1-4
Port Channel Port-Channel1:
  Active Ports: Ethernet17     Ethernet18     PeerEthernet1 PeerEthernet2
Port Channel Port-Channel2:
  Active Ports: Ethernet19     Ethernet20     Ethernet21     Ethernet22
                PeerEthernet3 PeerEthernet4 PeerEthernet5 PeerEthernet6
Port Channel Port-Channel3:
  Active Ports: Ethernet23     Ethernet24     PeerEthernet7 PeerEthernet8
Port Channel Port-Channel4:
  Active Ports: Ethernet25     Ethernet26     PeerEthernet9 PeerEthernet10
```

Issue the command **show port-channel detailed** command for channel 1 from Switch 2:

```
Switch#show port-channel 1 detailed
Port Channel Port-Channel1:
  Active Ports:
       Port                Time became active       Protocol    Mode
     ------------------------------------------------------------------------
       Ethernet17          7/7/11 15:27:36          LACP        Active
       Ethernet18          7/7/11 15:27:36          LACP        Active
       PeerEthernet1       7/7/11 15:27:36          LACP        Active
       PeerEthernet2       7/7/11 15:27:36          LACP        Active
```

12.5.5.5 Verify the VLAN Membership

The **show vlan** command displays VLAN member ports, including MLAG ports and ports on each peer not bundled in an MLAG.

```
Switch1#show vlan 3903, 3908
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
3903  ar.mg.rn.172.17.254.16/29        active    Cpu, Po1
3908  po.ra.ar.mg.172.17.254.64/29     active    Cpu, Et17, Et18, PEt17, PEt18
```

12.6 MLAG Commands

This section contains descriptions of the CLI commands that this chapter references.

MLAG and Port Channel Commands – Global Configuration Mode

- mlag configuration (global configuration) . . . . . Page 447

Interface Configuration Commands – Interface Configuration Mode

- mlag (port-channel interface configuration) . . . . . Page 446

MLAG Configuration Commands

- domain-id. . . . . Page 443
- heartbeat-interval . . . . . Page 444
- local-interface . . . . . Page 445
- peer-address . . . . . Page 448
- peer-link . . . . . Page 449
- reload-delay . . . . . Page 450
- shutdown (MLAG) . . . . . Page 454

Display Commands

- show mlag . . . . . Page 451
- show mlag interfaces . . . . . Page 453

domain-id

The **domain-id** command specifies a name for the Multichassis Link Aggregation (MLAG) domain.

The **no domain-id** and **default domain-id** commands remove the MLAG domain name by deleting the **domain-id** statement from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
domain-id identifier
no domain-id
default domain-id
```

Parameters

- *identifier* alphanumeric string that names the MLAG domain.

Examples

- This command names the MLAG domain ***mlag1***.

```
switch(config)#mlag
switch(config-mlag)#domain-id mlag1
switch(config-mlag)#
```

heartbeat-interval

The **heartbeat-interval** command configures the interval at which heartbeat messages are issued in a Multichassis Link Aggregation (MLAG) configuration.

The **no heartbeat-interval** and **default heartbeat-interval** commands revert the heartbeat interval to the default setting (2 seconds) by removing the **heartbeat-interval** command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
heartbeat-interval period
no heartbeat-interval
default heartbeat-interval
```

Parameters

- *period* Interval duration (ms). Value ranges from 1000 through 30000. Default interval is 2000 ms.

Guidelines

Heartbeat messages flow independently in both directions between the MLAG peers. If a peer stops receiving heartbeat messages within the expected time frame (2.5 times the heartbeat interval), the other peer can assume it no longer functions and without intervention or repair, the MLAG becomes disabled. Both switches revert to their independent state.

Examples

- This command configures the heartbeat interval to 15000 milliseconds:

```
switch(config)#mlag
switch(config-mlag)#heartbeat-interval 15000
switch(config-mlag)#
```

local-interface

The **local-interface** command assigns a VLAN interface for use in Multichassis Link Aggregation (MLAG) configurations. The VLAN interface is used for both directions of communication between the MLAG peers.

The **no local-interface** and **default local-interface** commands delete the VLAN interface assignment by removing the **local-interface** command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
local-interface vlan vlan_number
no local-interface
default local-interface
```

Parameters

- *vlan_number* VLAN number, in the range from 1 through 4094.

Guidelines

When configuring the local interface, the VLAN interface must exist already. To configure a VLAN interface, issue the command **interface vlan**.

Example

- This command assigns VLAN 4094 as the local interface.

```
switch(config)#mlag
switch(config-mlag)#local-interface vlan 4094
switch(config-mlag)#
```

mlag (port-channel interface configuration)

The **mlag** command assigns an MLAG ID to a port-channel. MLAG peer switches form an MLAG when each switch configures the same MLAG ID to a port-channel interface. Only one MLAG ID can be assigned to an interface. An individual MLAG number cannot be assigned to more than one interface.

The **no mlag** and **default mlag** commands remove the MLAG ID assignment from the configuration mode interface by deleting the corresponding **mlag** command from ***running-config***.

Command Mode

Interface-port-channel Configuration

Command Syntax

```
mlag number
no mlag
default mlag
```

Parameters

- *number* Number used as MLAG ID. Value ranges from 1 to 1000.

Example

- These commands configures a port channel and assigns it MLAG 4.

```
switch(config)#interface ethernet 5-10
switch(config-if-Et5-10)#channel-group 1 mode active
switch(config-if-Et5-10)#interface port-channel 4
switch(config-if-Po4)#switchport trunk group group4
switch(config-if-Po4)#mlag 4
switch(config-if-Po4)#exit
switch(config)#
```

mlag configuration (global configuration)

The **mlag configuration** command enters MLAG configuration mode to configure Multichassis Link Aggregation (MLAG) features. MLAG configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

The **no mlag configuration** and **default mlag configuration** commands remove all MLAG configuration commands from ***running-config***.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
mlag [configuration]
no mlag configuration
default mlag configuration
```

mlag and **mlag configuration** are identical commands.

Guidelines

An MLAG is formed by connecting two switches through an interface called a peer link. The peer link carries control and data traffic between the switches, including advertisements and keepalive messages. This information coordinates the switches. Functioning peers are in the **active** state.

Each peer switch uses IP-level connectivity between their local addresses and the MLAG peer IP address to form and maintain the peer link.

Commands Available in MLAG configuration mode

- **domain-id**
- **heartbeat-interval**
- **local-interface**
- **peer-address**
- **peer-link**
- **reload-delay**
- **shutdown (MLAG)**

Example

- These commands enter MLAG configuration mode and configure MLAG parameters:

```
switch(config)#mlag
switch(config-mlag)#local-interface vlan 4094
switch(config-mlag)#peer-address 10.0.0.2
switch(config-mlag)#peer-link port-channel 10
switch(config-mlag)#domain-id mlagDomain
switch(config-mlag)#heartbeat-interval 2500
switch(config-mlag)#reload-delay 2000
switch(config-mlag)#exit
switch(config)#
```

peer-address

The **peer-address** command specifies the peer IPv4 address for a Multichassis Link Aggregation (MLAG) domain. MLAG control traffic, including keepalive messages, is sent to the peer IPv4 address. If the peer IPv4 address is unreachable, then MLAG peering fails and both peer switches revert to their independent state.

The **no peer-address** and **default peer-address** commands remove the MLAG peer's IPv4 address assignment by deleting the peer-address command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
peer-address ipv4_addr
no peer-address
default peer-address
```

Parameters

- *ipv4_addr* MLAG peer IPv4 address.

Example

- These commands configure the MLAG peer address.

```
switch(config)#mlag
switch(config-mlag)#peer-address 10.0.0.2
switch(config-mlag)#
```

peer-link

The **peer-link** command specifies the interface that connects Multichassis Link Aggregation (MLAG) peers. To form an MLAG, two switches are connected through an interface called a peer link. The peer link carries control and data traffic between the two switches. Control traffic includes MLAG-related advertisements and keepalive messages. This information keeps the two switches working as one.

The **no peer-link** and **default peer-link** command remove the peer link by deleting the **peer-link** command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
peer-link int_name
no peer-link
default peer-link
```

Parameters

- *int_name* denotes the interface type and number of the interface. Values include:
  - — **ethernet** *e_num* Ethernet interface range specified by *e_num*.
  - — **port-channel** *p_num* Channel group interface range specified by *p_num*.

Example

- These commands creates a peer link.

```
switch(config)#mlag configuration
switch(config-mlag)#peer-link port-channel 10
switch(config-mlag)
```

reload-delay

The **reload-delay** command specifies the period that non-peer links are disabled after an MLAG peer reboots. This interval allows non-peer links to learn multicast and OSPF states before the ports start handling traffic. The recommended minimum value required to ensure the forwarding hardware is initialized with the topology state depends on the switch platform:

- fixed configuration switches: 300 seconds
- modular switches: 600 seconds

The **no reload-delay** and **default reload-delay** command restores the default value of 300 by deleting the **reload-delay** statement from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
reload-delay PERIOD
no reload-delay
default reload-delay
```

Parameters

- ***PERIOD*** Period that non-peer links are disbled after an MLAG peer reboots. Options include.
  - **infinity** link is not enabled after reboot.
  - *time* disabled link interval (seconds). Values range from 0 to 86400 (24 hours). Default is 300.

Example

- These commands configure the reload-delay interval to ten minutes.

```
switch(config)#mlag configuration
switch(config-mlag)#reload-delay 600
switch(config-mlag)#
```

show mlag

The **show mlag** command displays information about the Multichassis Link Aggregation (MLAG) configuration on bridged Ethernet interfaces.

Command Mode

EXEC

Command Syntax

```
show mlag [INFO_LEVEL]
```

Parameters

- *INFO_LEVEL* specifies information displayed by command. Options include:
  - <no parameter> command displays basic MLAG parameters.
  - **detail** command displays detailed MLAG interface parameters.

Display Values

Field names are listed in the order in which they appear in the output displays.

- **MLAG Configuration:**
  - ***domain-id*** Unique identifier used by peers for the MLAG domain.
  - ***local-interface*** VLAN interface configured to connect with MLAG peer.
  - ***peer-address*** Peer's IPv4 address for an MLAG domain.
  - ***peer-link*** Port Channel Interface that connects the MLAG peers.
- **MLAG Status**
  - ***Status*** Active, Inactive, Disabled.
  - ***peer-link status*** Unknown, Down, Up.
  - ***local-int status*** Up, Down, Testing, Unknown, Dormant, Not Present, LowerLayerDown.
  - **system-id** MAC address assigned to MLAG domain.
- **MLAG Ports**
  - ***Disabled*** Number of interfaces configured for MLAG that are disabled.
  - ***Configured*** Number of interfaces configured for MLAG.
  - ***Inactive*** Number of interfaces configured for MLAG that are inactive.
  - ***Active-Partial*** Number of active MLAG interfaces whose peers are inactive.
  - ***Active-Full*** Number of MLAG interfaces in active state with peer interfaces that are active.
- **MLAG Detailed Status**
  - ***State*** Internal state machine status. Primary, Secondary, Inactive, Disabled
  - ***State changes*** Number of state changes.
  - ***Last state change time*** Timestamp of the last state change.
  - ***Primary-priority*** Internal state machine variable.
  - ***Peer primary priority*** Internal state machine variable of the MLAG peer.
  - ***Peer MAC address*** MAC address of the MLAG peer.
  - ***Recently rebooted*** Whether the switch has recently rebooted. Values are True or False.
  - ***Last recently rebooted change time*** Timestamp of the last switch reboot.
  - ***State decided by recently rebooted*** State of peer renegotiation following reboot. True, False.
  - ***Heartbeat-interval*** Period between keepalive messages (1000 to 30000 ms). Default is 5000 ms.
  - ***heartbeat-timeout*** Period after keepalive message until MLAG is disabled.
  - ***Agent should be running*** True, False.

Examples

- This command displays output from the **show mlag** command:

```
switch>show mlag
MLAG Configuration:
domain-id          :          ar.mg.mlag
local-interface    :            Vlan3901
peer-address       :        172.17.254.2
peer-link          :       Port-Channel1

MLAG Status:
state              :              Active
peer-link status   :                  Up
local-int status   :                  Up
system-id          :   02:1c:73:00:13:19

MLAG Ports:
Disabled           :   0
Configured         :   0
Inactive           :   0
Active-partial     :   0
Active-full        :   5
switch>
```

show mlag interfaces

The **show mlag interfaces** command displays information about the Multichassis Link Aggregation (MLAG) configuration on bridged Ethernet interfaces.

Command Mode

EXEC

Command Syntax

```
show mlag interfaces [MLAGS] [INFO_LEVEL]
```

Parameters

- *MLAGS* MLAG channels for which command displays data. Options include:
  - <no parameter> command displays data for all MLAGs.
  - *mlag_id* specifies MLAG for which command displays data. Value ranges from 1 to 1000.
- *INFO_LEVEL* specifies information displayed by command. Options include:
  - <no parameter> command displays basic MLAG interface parameters
  - **detail** command displays detailed MLAG interface parameters.

Display Values

Field names are listed in the order in which they appear in the output displays.

- **Basic Interface Parameters**

- *MLAG* MLAG number assigned to interface.
- *Desc* Description of the Port Channel interface.
- *State* Activity level of interface.
- *local* Port Channel Interface number.
- *remote* Port Channel number of peer interface.
- *local/remote status* status of MLAG port and peer.

- **Detailed Interface Parameters**

- *MLAG* MLAG number assigned to interface.
- *State* Activity level of interface.
- *local* Port Channel Interface number.
- *remote* Port Channel number of peer interface.
- *local/remote status* status of MLAG port and peer.
- *local/remote config* configuration status of MLAG port and peer.
- *last change* elapsed time since last change to interface.
- *changes* number of changes to interface.

Example

- This command displays output from the **show mlag interfaces detail** command:

```
switch>show mlag interfaces detail
                                  local/remote
mlag       state local remote   oper   config          last change  changes
--------------------------------------------------------------------------------
   4 active-full   Po4    Po4  up/up  ena/ena  6 days, 1:19:26 ago        5
   5 active-full   Po5    Po5  up/up  ena/ena  6 days, 1:19:24 ago        5
   6 active-full   Po6    Po6  up/up  ena/ena  6 days, 1:19:23 ago        5
   7 active-full   Po7    Po7  up/up  ena/ena  6 days, 1:19:23 ago        5
```

shutdown (MLAG)

The **shutdown** command disables MLAG on the switch without modifying the MLAG configuration.

The **no shutdown** and **default shutdown** commands re-enable MLAG by removing the **shutdown** command from ***running-config***.

Command Mode

MLAG Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Example

- These commands disable MLAG on the switch.

```
switch(config)#mlag configuration
switch(config-mlag)#shutdown
switch(config-mlag)#
```

Chapter 13

ACLs and Route Maps

The switch uses rule based lists to control packet access to ports and to select routes for redistribution to routing domains defined by dynamic routing protocols. This chapter describes the construction of Access Control Lists (ACLs) and route maps.

This chapter includes the following sections:

- Section 13.1: ACL, Route Map, and Prefix List Introduction
- Section 13.2: Access Control Lists
- Section 13.3: Route Maps
- Section 13.4: Prefix Lists
- Section 13.5: ACL, Route Map, and Prefix List Commands

13.1 ACL, Route Map, and Prefix List Introduction

An access control list (ACL) is an ordered set of rules that control the inbound flow of packets into Ethernet interfaces, port channel interfaces or the switch control plane. The switch supports the implementation of a wide variety of filtering criteria including IP and MAC addresses, TCP/UDP ports with include/exclude options without compromising its performance or feature set. Filtering syntax is industry standard.

A route map is an ordered set of rules that control the redistribution of IP routes into a protocol domain on the basis of such criteria as route metrics, access control lists, next hop addresses, and route tags. Route maps can also alter parameters of routes as they are redistributed.

A prefix list is an ordered set of rules that defines route redistribution access for a specified IP address space. Route maps often use prefix lists to filter routes.

13.2 Access Control Lists

These sections describe Access Control Lists:

- Section 13.2.1: ACL Description
- Section 13.3: Route Maps
- Section 13.4: Prefix Lists

13.2.1 ACL Description

This section describes ACL composition and function. The switch support the following ACL types:

- IPv4
- IPv6
- Standard IPv4
- Standard IPv6
- MAC

13.2.1.1 ACL Structure

An ACL is an ordered list of rules that defines access restiction for entities (interface or the control plane) to which is is applied. ACLs are also used by route maps to select routes for redistribution into specified routing domains.

ACL rules specify the data to which packet contents are compared when filtering data.

- The interface forwards packets that match all conditions in a permit rule.
- The interface drops packets that match all conditions in a deny rule.
- The interface drops packets that do not match at least one rule.

Upon its arrival at an interface, a packet's fields are compared to the first rule of the ACL applied to the interface. Packets that match the rule are forwarded (permit rule) or dropped (deny rule). Packets that do not match the rule are compared to the next rule in the list. This process continues until the packet either matches a rule or the rule list is exhausted. The interface drops packets not matching a rule.

The sequence number designates the rule's placement in the ACL.

13.2.1.2 ACL Rules

ACL rules consist of a condition list that is compared to inbound packet fields. When all of a rule's criteria match a packet's contents, the interface performs the action specified by the rule.

The set of available conditions depend on the ACL type and the specified protocol within rule. The following is a list of conditions available for supported ACL types

IPv4 ACL Rule Parameters

All rules in IPv4 ACLs include the following criteria:

- **Protocol:** The packet's IP protocol. Valid rule inputs include:
  - Protocol name for a limited set of common protocols.
  - Assigned protocol number for all IP protocols.
- **Source Address:** The packet's source IPv4 address. Valid rule inputs include:
  - a subnet address (CIDR or address-mask). Discontiguous masks are supported.
  - a host IP address (dotted decimal notation).
  - ***any*** to denote that the rule matches all source addresses.

- **Destination Address:** The packet's destination IP address. Valid rule inputs include:
  - a subnet address (CIDR or address-mask). Discontiguous masks are supported.
  - a host IP address (dotted decimal notation).
  - ***any*** to denote that the rule matches all destination addresses.

All rules in IPv4 ACLs ***may*** include the following criteria:

- **Fragment:** Rules filter on the fragment bit.
- **Time-to-live:** Compares the TTL (time-to-live) value in the packet to a specified value. Valid in ACLs applied to the Control Plane. Validity in ACLs applied to the data plane varies by switch platform. Comparison options include:
  - ***Equal:*** Packets match if packet value equals statement value.
  - ***Greater than:*** Packets match if packet value is greater than statement value.
  - ***Less than:*** Packets match if packet value is less than statement value.
  - ***Not equal:*** Packets match if packet value does not equals statement value.

The availability of the following optional criteria depends on the specified protocol:

- **Source Ports / Destination Ports:** A rule filters on ports when the specified protocol supports IP address-port combinations. Rules provide one of these port filtering values:
  - ***any*** denotes that the rule matches all ports.
  - A list of ports that matches the packet port. Maximum list size is 10 ports
  - Negative port list. The rule matches any port not in the list. Maximum list size is 10 ports.
  - Integer (lower bound): The rule matches any port with a number larger than the integer.
  - Integer (upper bound): The rule matches any port with a number smaller than the integer.
  - Range integers: The rule matches any port whose number is between the integers.
- **Flag bits**: Rules filter TCP packets on flag bits.
- **Message type**: Rules filter ICMP type or code.
- **Tracked:** Matches packets in existing ICMP, UDP, or TCP connections. Valid in ACLs applied to the Control Plane. Validity in ACLs applied to the data plane varies by switch platform.

IPv6 ACL Rule Parameters

All rules in IPv6 ACLs include the following criteria:

- **Protocol:** All rules filter on the packet's IP protocol field. Rule input options include:
  - Protocol name for a limited set of common protocols.
  - Assigned protocol number for all IP protocols.
- **Source Address:** The packet's source IPv6 address. Valid rule inputs include:
  - a IPv6 prefix (CIDR). Discontiguous masks are supported.
  - a host IP address (dotted decimal notation).
  - ***any*** to denote that the rule matches all addresses.
- **Destination Address:** The packet's destination IP address. Valid rule inputs include:
  - a subnet address (CIDR or address-mask). Discontiguous masks are supported.
  - a host IP address (dotted decimal notation).
  - ***any*** to denote that the rule matches all addresses.

All rules in IPv6 ACLs ***may*** include the following criteria:

- **Fragment:** Rules filter on the fragment bit.

- ***HOP*** Compares the packet's hop-limit value to a specified value. Comparison options include:
  - ***Equal:*** Packets match if packet value equals statement value.

The availability of the following optional criteria depends on the specified protocol:

- **Source Ports / Destination Ports:** A rule filters on ports when the specified protocol supports IP address-port combinations. Rules provide one of these port filtering values:
  - ***any*** denotes that the rule matches all ports.
  - A list of ports that matches the packet port. Maximum list size is 10 ports
  - Negative port list. The rule matches any port not in the list. Maximum list size is 10 ports.
  - Integer (lower bound): The rule matches any port with a number larger than the integer.
  - Integer (upper bound): The rule matches any port with a number smaller than the integer.
  - Range integers: The rule matches any port whose number is between the integers.
- **Flag bits**: Rules filter TCP packets on flag bits.
- **Message type**: Rules filter ICMP type or code.
- **Tracked:** Matches packets in existing ICMP, UDP, or TCP connections. Valid in ACLs applied to the Control Plane. Validity in ACLs applied to the data plane varies by switch platform.

Standard IPv4 and IPv6 ACL Rule Parameters

Standard ACLs filter only on the source address.

MAC ACL Rule Parameters

MAC ACLs filter traffic on a packet's layer 2 header. Criteria that MAC ACLs use to filter packets include:

- **Source Address** and **Mask:** The packet's source MAC address. Valid rule inputs include:
  - MAC address range (address-mask in 3x4 dotted hexadecimal notation).
  - ***any*** to denote that the rule matches all source addresses.
- **Destination Address** and **Mask:** The packet's destination MAC address. Valid rule inputs include:
  - MAC address range (address-mask in 3x4 dotted hexadecimal notation).
  - ***any*** to denote that the rule matches all destination addresses.
- **Protocol:** The packet's protocol as specified by its EtherType field contents. Valid inputs include:
  - Protocol name for a limited set of common protocols.
  - Assigned protocol number for all protocols.

13.2.1.3 Creating and Modifying Lists

The switch provides configuration modes for creating and modifying ACLs. The command that enters an ACL Configuration mode specifies the name of the list that the mode modifies. The switch saves the list to the running configuration when the configuration mode is exited.

- ACLs are created and modified in ACL Configuration mode.
- Standard ACLs are created and modified in Standard-ACL-Configuration mode.
- MAC ACLs are created and modified in MAC-ACL-Configuration mode.

Lists that are created in one mode cannot be modified in any other mode.

A sequence number designates the rule's placement in a list. New rules are inserted into a list according to their sequence numbers. A rule's sequence number can be referenced when deleting it from a list.

Section 13.2.2 describes procedures for configuring ACLs.

13.2.1.4 Implementing Access Control Lists

An access control list is implemented by assigning the list to an Ethernet or Port Channel interface, or to the Control Plane. The switch assigns a default ACL to the Control Plane unless the configuration contains a valid Control-Plane ACL assignment statement. Ethernet and Port Channel interfaces are not assigned an ACL by default. Standard ACLs are applied to interfaces in the same manner as other ACLs.

IPv4 and MAC ACLs are separately applied for inbound and outbound packets. An interface can be assigned multiple ACLs, with a limit of one ACL per packet direction per ACL type. Egress ACLs are supported on a subset of all available switches. The control-plane does not support egress ACLs.

Section 13.2.3 describes procedures for applying ACLs to interfaces or the control plane.

13.2.1.5 ACL Rule Tracking

ACL rule tracking determines the impact of ACL rules on the traffic accessing interfaces upon which they are applied. ACLs provide two tracking mechanisms:

- ACL logging: A syslog entry is logged when a packet matches specified ACL rules.
- ACL counters: ACL counters increment when a packet matches a rule in specified ACLs.

ACL Logging

ACL rules provide a **log** option that produces a log message when a packet matches the rule. ACL logging creates a syslog entry when a packet matches an ACL rule where logging is enabled. Packets that match a logging-enabled ACL rule is copied to the CPU by the hardware. These packets trigger the creation of a syslog entry. The information provided by entry depend on the ACL type or the protocol specified by the ACL. Hardware rate limiting is applied to packets written to the CPU, avoiding potential DoS attacks. The rate of logging is also software limited to avoid the creation of syslog lists that are too large for practical use by human operators.

Section 13.2.2.3 describes procedures for configuring and enabling ACL logging.

ACL Counters

An ACL counter is assigned to each ACL rule. The activity of the ACL counters for rules within a list depend on the list's counter state. When the list is in counting state, the ACL counter of a rule increments when the rule matches a packet. When the list is in non-counting state, the counter does not increment. A list's counter state applies to all rules in the ACL. The initial state for new ACLs is non-counting.

When an ACL changes from counting state to non-counting state, or when the ACL is no longer applied to any interfaces that increment counters, counters for all rules in the list maintain their values and do not reset. When the ACL returns to counting mode or is applied to an interface that increments counters, the counter operation resumes from its most recent value.

Counters never decrement and are reset only through CLI commands.

Section 13.2.2.3 describes procedures for configuring and enabling ACL counters.

13.2.2 ACL Configuration

Access Control Lists are created and modified in an ACL-configuration mode. A list can be edited only in the mode where it was created.The switch provides five configuration modes for creating and modifying Access Control Lists:

- **ACL configuration Mode** for IPv4 Access Control Lists.
- **IPv6-ACL configuration Mode** for IPv6 Access Control Lists.
- **Std-ACL configuration Mode** for Standard IPv4 Access Control Lists.
- **Std-IPv6-ACL configuration Mode** for Standard IPv6 Access Control Lists.
- **MAC-ACL configuration Mode** for MAC Access Control Lists.

These sections describe the creation and modification of ACLs:

- Section 13.2.2.1: Managing ACLs
- Section 13.2.2.2: Modifying an ACL
- Section 13.2.2.3: ACL Rule Tracking Configuration
- Section 13.2.2.4: Displaying ACLs

13.2.2.1 Managing ACLs

Creating and Opening a List

To create an ACL, enter one of the following commands, followed by the name of the list:

- **ip access-list** for IPv4 ACLs.
- **ipv6 access-list** for IPv6 ACLs.
- **ip access-list standard** for standard IPv4 ACLs.
- **ipv6 access-list standard** for standard IPv6 ACLs.
- **mac access-list** for MAC ACLs.

The switch enters the appropriate ACL configuration mode for the list. If the command is followed by the name of an existing ACL, subsequent commands edit that list.

Examples

- This command places the switch in ACL configuration mode to create an ACL named ***test1***.

```
switch(config)#ip access-list test1
switch(config-acl-test1)#
```

- This command places the switch in Standard-ACL-Configuration mode to create a Standard ACL named ***stest1***.

```
switch(config)#ip access-list standard stest1
switch(config-std-acl-stest1)#
```

- This command places the switch in MAC-ACL configuration mode to create an MAC ACL named ***mtest1***.

```
switch(config)#mac access-list mtest1
switch(config-mac-acl-mtest1)#
```

Saving List Modifications

ACL configuration modes are group-change modes. Changes made in a group-change mode are saved by exiting the mode.

Important After exiting ACL mode, the running-config file must be saved to the startup configuration file to preserve an ACL after a system restart.

Example

- The second example in Adding a Rule (page 462) results in this edited ACL:

```
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.30.10.0/24 host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Because the changes were not yet saved, the ACL remains empty, as shown by **show ip access-lists**.

```
switch(config-acl-test1)#show ip access-lists test1
switch(config-acl-test1)#
```

To save all current changes to the ACL and exit ACL edit mode, type **exit** at the prompt.

```
switch(config-acl-test1)#exit
switch(config)#show ip access-lists test1
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.30.10.0/24 host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Discarding List Changes

The **abort** command exits ACL edit mode without saving pending changes.

Example

- Example 2 in Adding a Rule (page 462) results in this edited ACL:

```
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.30.10.0/24 host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

To discard the changes, enter **abort**. If the ACL existed before entering ACL-Configuration Mode, **abort** restores the version that existed before entering ACL-Configuration Mode. Otherwise, **show ip access-lists** shows the ACL was not created.

```
switch(config-acl-test1)#abort
switch(config)#
```

13.2.2.2 Modifying an ACL

These commands add deny rules to the appropriate ACL:

- **deny (IPv4 ACL)** adds a deny rule to an IPv4 ACL.
- **deny (IPv6 ACL)** adds a deny rule to an IPv6 ACL.
- **deny (Standard IPv4 ACL)** adds a deny rule to an IPv4 standard ACL.
- **deny (Standard IPv6 ACL)** adds a deny rule to an IPv6 standard ACL.
- **deny (MAC ACL)** adds a deny rule to a MAC ACL.

These commands add permit rules to the appropriate ACL:

- **permit (IPv4 ACL)** adds a permit rule to an IPv4 ACL.
- **permit (IPv6 ACL)** adds a permit rule to an IPv6 ACL.
- **permit (Standard IPv4 ACL)** adds a permit rule to an IPv4 standard ACL.
- **permit (Standard IPv6 ACL)** adds a permit rule to an IPv6 standard ACL.
- **permit (MAC ACL)** adds a permit rule to a MAC ACL.

Adding a Rule

To append a rule to the end of a list, enter the rule without a sequence number while in ACL Configuration mode for the list. The new rule's sequence number is derived by adding 10 to the last rule's sequence number.

Examples

- These commands enter the first three rules into a new ACL.

```
switch(config-acl-test1)#permit ip 10.10.10.0/24 any
switch(config-acl-test1)#permit ip any host 10.20.10.1
switch(config-acl-test1)#deny ip host 10.10.10.1 host 10.20.10.1
```

  To view the edited list, type **show**.

```
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
```

- This command appends a rule to the active ACL. The sequence number of new rule is 40.

```
switch(config-acl-test1)#permit ip any any
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Inserting a Rule

To insert a rule into a ACL, enter the rule with a sequence number between the existing rules' numbers.

Example

- This command inserts a rule between the first two rules by assigning it the sequence number 15.

```
Switch(config-acl-test1)#15 permit ip 10.30.10.0/24 host 10.20.10.1
Switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        15 permit ip 10.30.10.0/24 host 10.20.10.1
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Deleting a Rule

To remove a rule from the current ACL, perform one of these commands:

- Enter **no**, followed by the sequence number of the rule to be deleted.
- Enter **no**, followed by the rule be deleted.
- Enter **default**, followed by the rule to be deleted.

Example

- These equivalent commands remove rule 20 from the list.

```
switch(config-acl-test1)#no 20

switch(config-acl-test1)#no permit ip any host 10.20.10.1

switch(config-acl-test1)#default permit ip any host 10.20.10.1
```

This ACL results from entering one of the preceding commands.

```
Switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        15 permit ip 10.30.10.0/24 host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
```

Resequencing Rule Numbers

Sequence numbers determine the order of the rules in an Access Control List. After a list editing session where existing rules are deleted and new rules are inserted between existing rules, the sequence number distribution may not be uniform. Resequencing rule numbers changes adjusts the sequence number of rules to provide a constant difference between adjacent rules. The **resequence** command adjusts the sequence numbers of ACL rules.

Example

- The **resequence** command renumbers rules in the test1 ACL. The sequence number of the first rule is 100; subsequent rules numbers are incremented by 20.

```
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        25 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        50 permit ip any any
        90 remark end of list
switch(config-acl-test1)#resequence 100 20
switch(config-acl-test1)#show
IP Access List test1
        100 permit ip 10.10.10.0/24 any
        120 permit ip any host 10.20.10.1
        140 deny ip host 10.10.10.1 host 10.20.10.1
        160 permit ip any any
        180 remark end of list
```

13.2.2.3 ACL Rule Tracking Configuration

ACL rules provide a **log** option that produces a syslog message about the packets matching packet. ACL logging creates a syslog entry when a packet matches an ACL rule with logging enabled.

Example

- This command creates an ACL rule with logging enabled.

```
switch(config-acl-test1)#15 permit ip 10.30.10.0/24 host 10.20.10.1 log
switch(config-acl-test1)#
```

The format of the generated syslog message depends on the ACL type and the specified protocol:

- Messages generated by a TCP or UDP packet matching an IP ACL use the following format:

  IPACCESS: list *acl intf filter protocol src-ip*(*src_port*) **->** *dst-ip*(*dst_port*)

- Messages generated by ICMP packets matching an IP ACL use this format:

  IPACCESS: list *acl intf filter* **icmp** *src-ip*(*src-port*) **->** *dst-ip*(*dst-port*) **type**=*n* **code**=*m*

- Messages generated by all other IP packets matching an IP ACL use this format:

  IPACCESS: list *acl intf filter protocol src-ip* **->** *dst-ip*

- Messages generated by packets matching a MAC ACL use this format:

  MACACCESS: list *acl intf filter vlan ether src_mac* **->** *dst_mac*

- Messages generated by a TCP or UDP packet matching a MAC ACL use this format:

  MACACCESS: list *acl intf filter vlan ether ip-prt src-mac src-ip*:*src-prt* **->** *dst-mac dst-ip*:*dst-prt*

- Messages generated by any other IP packet matching a MAC ACL use this format:

  MACACCESS: list *acl intf filter vlan ether src_mac src_ip* **->** *dst_mac dst_ip*

Variables in the syslog messages display the following values:

- *acl* Name of ACL.
- *intf* Name of interface that received the packet.
- *filter* Action triggered by ACL (**denied** or **permitted**).
- *protocol* IP protocol specified by packet.

- *vlan* Number of VLAN receiving packet.
- *ether* Ethertype protocol specified by packet.
- *src-ip* and *dst-ip* source and destination IP addresses.
- *src-prt* and *dst-prt* source and destination ports.
- *src-mac* and *dst-mac* source and destination MAC addresses.

ACLs provide a command that configures its counter state (counting or non-counting). The counter state applies to all rules in the ACL. The initial state for new ACLs is non-counting.

The **statistics per-entry (ACL configuration modes)** command places the ACL in counting mode.

- This command places the configuration mode ACL in counting mode.

```
switch(config-acl-test1)#statistics per-entry
switch(config-acl-test1)#exit
switch(config-acl-test1)#show ip access-list test1
IP Access List test1
        statistics per-entry
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
```

The **clear ip access-lists counters** and **clear ipv6 access-lists counters** commands set the IP access list counters to zero for the specified IP access list.

- This command clears the ACL counter for the test1 ACL.

```
switch(config)#clear ip access-lists counters test1
switch(config)#
```

13.2.2.4 Displaying ACLs

ACLs can be displayed by a **show running-config** command. The **show ip access-lists** also displays ACL rosters and contents, as specified by command parameters.

When editing an ACL, the **show (ACL configuration modes)** command displays the current or pending list, as specified by command parameters.

Displaying a List of ACLs

To display the roster of ACLs on the switch, enter **show ip access-lists** with the **summary** option.

Example

- This command lists the available Access Control Lists.

```
switch(config)#show ip access-list summary
IPV4 ACL default-control-plane-acl
        Total rules configured: 12
        Configured on: control-plane
        Active on    : control-plane

IPV4 ACL list2
        Total rules configured: 3

IPV4 ACL test1
        Total rules configured: 6
```

```
IPV4 ACL test_1
        Total rules configured: 1

IPV4 ACL test_3
        Total rules configured: 0

switch(config)#
```

Displaying Contents of an ACL

These commands display ACL contents.

- **show ip access-lists**
- **show ipv6 access-lists**
- **show mac access-lists**

Each command can display the contents of one ACL or of all ACLs of the type specified by the command:

- To display the contents of one ACL, enter **show ip access-lists** followed by the name of the ACL.
- To display the contents of all ACLs on the switch, enter the command without any options.

ACLs that are in counting mode display the number of inbound packets each rule in the list matched and the elapsed time since the last match.

Example

- This command displays the rules in the ***default-control-plane-acl*** ACL.

```
switch#show ip access-lists default-control-plane-acl
IP Access List default-control-plane-acl [readonly]
        statistics per-entry
        10 permit icmp any any
        20 permit ip any any tracked [match 1725, 0:00:00 ago]
        30 permit ospf any any
        40 permit tcp any any eq ssh telnet www snmp bgp https
        50 permit udp any any eq bootps bootpc snmp [match 993, 0:00:29 ago]
        60 permit tcp any any eq mlag ttl eq 255
        70 permit udp any any eq mlag ttl eq 255
        80 permit vrrp any any
        90 permit ahp any any
        100 permit pim any any
        110 permit igmp any any [match 1316, 0:00:23 ago]
        120 permit tcp any any range 5900 5910
```

- This command displays the rules in all ACLs on the switch.

```
switch#show ip access-lists
IP Access List default-control-plane-acl [readonly]
        statistics per-entry
        10 permit icmp any any
        20 permit ip any any tracked [match 1371, 0:00:00 ago]
        30 permit ospf any any
        40 permit tcp any any eq ssh telnet www snmp bgp https
        50 permit udp any any eq bootps bootpc snmp
        60 permit tcp any any eq mlag ttl eq 255
        70 permit udp any any eq mlag ttl eq 255
        80 permit vrrp any any
        90 permit ahp any any
        100 permit pim any any
        110 permit igmp any any [match 1316, 0:00:23 ago]
        120 permit tcp any any range 5900 5910

IP Access List list2
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.30.10.0/24 host 10.20.10.1
        30 permit ip any host 10.20.10.1
        40 deny ip host 10.10.10.1 host 10.20.10.1
        50 permit ip any any

IP Access List test1
<-------OUTPUT OMITTED FROM EXAMPLE-------->

Switch(config)#
```

Displaying ACL Modifications

While editing an ACL in ACL-Configuration mode, the **show (ACL configuration modes)** command provides options for displaying ACL contents.

- To display the list, as modified in ACL configuration mode, enter **show** or **show pending**.
- To display the list, as stored in *running-config*, enter **show active**.
- To display differences between the pending list and the stored list, enter **show diff**.

Examples

The examples in this section assume these ACL commands were previously entered.

These commands are stored in the configuration:

```
10 permit ip 10.10.10.0/24 any
20 permit ip any host 10.21.10.1
30 deny ip host 10.10.10.1 host 10.20.10.1
40 permit ip any any
50 remark end of list
```

The current edit session removed this command. This change is not yet stored to running-config:

```
20 permit ip any host 10.21.10.1
```

The current edit session added these commands ACL. They are not yet stored to running-config:

```
20 permit ip 10.10.0.0/16 any
25 permit tcp 10.10.20.0/24 any
45 deny pim 239.24.124.0/24 10.5.8.4/30
```

- This command displays the pending ACL, as modified in ACL Configuration Mode.

```
switch(config-acl-test_1)#show pending
IP Access List test_1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.10.0.0/16 any
        25 permit tcp 10.10.20.0/24 any
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        45 deny pim 239.24.124.0/24 10.5.8.4/30
        50 remark end of list
```

- This command displays the ACL, as stored in the configuration

```
switch(config-acl-test_1)#show active
IP Access List test_1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.21.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
```

- This command displays the difference between the saved and modified ACLs.
  - — Rules added to the pending list are denoted with a plus sign (+).
  - — Rules removed from the saved list are denoted with a minus sign (-).

```
switch(config-acl-test_1)#show diff
---
+++
@@ -1,7 +1,9 @@
 IP Access List test_1
         10 permit ip 10.10.10.0/24 any
-        20 permit ip any host 10.21.10.1
+        20 permit ip 10.10.0.0/16 any
+        25 permit tcp 10.10.20.0/24 any
         30 deny ip host 10.10.10.1 host 10.20.10.1
         40 permit ip any any
+        45 deny pim 239.24.124.0/24 10.5.8.4/30
```

13.2.3 Applying ACLs

Access Control Lists become active when they are assigned to an interface or the Control Plane. This section describes the process of adding and removing ACL interface assignments. Section 10.5.3 describes the process of applying an ACL to the control plane.

Applying an Access Control List to an Interface

The switch must be in interface configuration mode to assign an ACL to an interface.

- The **ip access-group (Interface Configuration modes)** command applies the specified IP or standard IP ACL to the configuration mode interface.
- The **mac access-group** command applies the specified MAC ACL to the configuration mode interface.

IPv4, IPv6, and MAC ACLs are separately applied for inbound and outbound packets. An interface can be assigned multiple ACLs, with a limit of one ACL per packet direction per ACL type. Egress ACLs are supported on a subset of all available switches. IPv6 egress ACLs have limited availability.

Example

- These commands assign ***test1*** ACL to Ethernet 3 interface, then verifies the assignment.

```
switch(config)#interface ethernet 3
switch(config-if-Et3)#ip access-group test1 in
switch(config-if-Et3)#show running-config interfaces ethernet 3
interface Ethernet3
   ip access-group test1 in
switch(config-if-Et3)#
```

Removing an ACL from an Interface

The **no ip access-group** command removes an IP ACL assignment statement from ***running-config*** for the configuration mode interface. After an ACL is removed, the interface is not associated with an IP ACL.

The **no mac ip access-group** command removes a MAC ACL assignment statement from ***running-config*** for the configuration mode interface. After a MAC ACL is removed, the interface is not associated with an MAC ACL.

To remove an ACL from the control plane, enter the **no ip access-group** command in control plane configuration mode. Removing the control plane ACL command from ***running-config*** reinstates ***default-control-plane-acl*** as the control plane ACL.

Examples

- These commands remove the assigned IPv4 ACL from Ethernet 3 interface.

```
switch(config)#interface ethernet 3
switch(config-if-Et3)#no ip access-group test in
switch(config-if-Et3)#
```

- These commands place the switch in control plane configuration mode and remove the ACL assignment from ***running-config***, restoring ***default-control-plane-acl*** as the Control Place ACL.

```
switch(config)#control-plane
switch(config-cp)#no ip access-group test_cp in
switch(config-cp)#
```

13.3 Route Maps

A route map is an ordered set of rules that control the redistribution of IP routes into a protocol domain on the basis of such criteria as route metrics, access control lists, next hop addresses, and route tags. Route maps can also alter parameters of routes as they are redistributed.

These sections describe the route map implementation:

- Section 13.3.1 describes route maps.
- Section 13.3.2 describes the route map configuration process.
- Section 13.3.3 describes the usage of route maps.

13.3.1 Route Map Description

Route maps are composed of route map clauses, each of which consists of a list of match and set statements.

Route Map Clauses

Route map clauses are categorized by the resolution of routes that the clause filters.

- Permit clauses facilitate the redistribution of matched routes.
- Deny clauses prevent the redistribution of matched routes.

Route map clause elements include name, sequence number, filter type, match statements, set statements, and continue statements.

- **name** identifies the route map to which the clause belongs.
- **sequence number** designates the clause's placement within the route map.
- **filter type** specifies the route resolution. Valid types are ***permit*** and ***deny***.
- **match statements** specify criteria that select routes that the clause is evaluating for redistribution.
- **set statements** modify route parameters for redistributed routes.
- **continue statements** prolong the route map evaluation of routes that match a clause.

Clauses filter routes for redistribution. Routes tha clauses pass are redistributed (permit clauses) or rejected (deny clauses). Routes that clauses fail are filtered by the next clause in the route map.

- When a clause does not contain a **match** statement, the clause passes all routes.
- When a clause contains a single **match** statement that lists a single object, the clause passes routes whose parameters match the object.
- When a clause contains a single **match** statement that lists multiple objects, the clauses passes routes whose parameters match at least one object.
- When a clause contains multiple **match** statements, the clause passes routes whose parameters match all match statements.

Set statements modify parameters for redistributed routes. Set statements are valid in permit clauses.

Example

The following route map clause is named MAP_1 with sequence number 10. The clause matches all routes from BGP Autonomous system 10 and redistributes them with a local preference set to 100. Routes that do not match the clause are evaluated against the next clause in the route map.

```
route-map MAP_1 permit 10
   match as 10
   set local-preference 100
```

Route Maps with Multiple Clauses

A route map consists of clauses with the same name and different sequence numbers. Clauses filter routes in ascending order of their sequence numbers. When a clause passes a route, the redistribution action is performed as specified by the filter type and all subsequent clauses are ignored. When the clause fails the route, the clause with the next lowest sequence number filters the route.

All route maps have an implied final clause that contains a single deny clause with no match statement. This denies redistribution to routes that are not passed by any clause.

Example

The following route map is named MAP_1 with two permit clauses. Routes that do not match either clause are denied redistribution into the target protocol domain.

```
route-map MAP_1 permit 10
   match as 10
   set local-preference 100
!
route-map MAP_1 permit 20
   match metric-type type-1
   match as 100
```

Section 13.3.2 describes route map configuration procedures.

Route Maps with Multiple Clauses and Continue Statements

Route map clauses that contain a **continue (route-map configuration mode)** command support additional route map evaluation of routes whose parameters meet the clauses's matching statements. Routes that match a clause containing a **continue** statement are evaluated against the clause specified by the **continue** statement.

When a route matches multiple route-map clauses, the filter action (deny or permit) is determined by the last clause that the route matches. The **set** statements in all clauses matching the route are applied to the route after the route map evaluation is complete. Multiple set statements are applied in the same order by which the route was evaluated against the clauses containing them.

Example

- The following route map is named MAP_1 with a permit clause and a deny clause. The permit clause contains a continue statement. Routes that map clause 10 are evaluated against clause 20.

```
route-map MAP_2 permit 10
   match as 10
   continue 20
   set local-preference 100
!
route-map MAP_2 deny 20
   match metric-type type-1
   match as 100
```

The route is redistributed if it passes clause 10 and is rejected by by clause 20. The route is denied redistribution in all other instances. The **continue** statement guarantees the evaluation of all routes against both clauses.

13.3.2 Route Map Configuration

Route maps are created and modified in route map configuration mode. These sections describe the configuration mode and its commands.

- Section 13.3.2.1: Route Map Creation and Editing
- Section 13.3.2.2: Modifying Route Map Components

13.3.2.1 Route Map Creation and Editing

Creating a Route Map Clause

To create a route map, enter **route-map** followed by the map name and filter type (**deny** or **permit**). The default sequence number is assigned to the clause if the command does not include a number.

Example

- This command places the switch in route map configuration mode to create a route map clause named ***map1*** with a sequence number of 50.

```
switch(config)#route-map map1 permit 50
switch(config-route-map-map1)#
```

Editing a Route Map Clause

To edit an existing route map clause, enter **route-map** with the map's name and clause's number. The switch enters route-map configuration mode for the clause. Subsequent **match (route-map configuration mode)** and **set (route-map configuration mode)** commands add the corresponding statements to the clause.

The **show (route-map configuration mode)** command displays contents of the existing route map.

Example

- This command places the switch in route map configuration mode to edit an existing route map clause. The **show** command displays contents of all clauses in the route map.

```
switch(config)#route-map MAP2
switch(config-route-map-MAP2)#show
route-map MAP2 deny 10
  Match clauses:
    match as 10
    match tag 333
  Set clauses:
    set local-preference 100
switch(config-route-map-MAP2)#
```

Saving Route Map Modifications

Route map configuration mode is a group-change mode. Changes are saved by exiting the mode.

Example

- The first command creates the ***map1*** clause with sequence number of 10. The second command is not yet saved to the route map, as displayed by the **show** command.

```
switch(config)#route-map map1 permit
switch(config-route-map-map1)#match as 100
switch(config-route-map-map1)#show


switch(config-route-map-map1)#
```

The **exit** command saves the **match** command.

```
switch(config-route-map-map1)#exit
switch(config)#show route-map map1
route-map map1 permit 10
  Match clauses:
    match as 100
  Set clauses:
switch(config)#
```

Discarding Route Map Modifications

The **abort** command discards all pending changes and exits route map configuration mode.

Example

The **abort** command discards the pending **match** command and restores the original route map.

```
switch(config)#route-map map1 permit
switch(config-route-map-map1)#match as 100
switch(config-route-map-map1)#abort
switch(config)#show route-map map1
switch(config)#
```

13.3.2.2 Modifying Route Map Components

These commands add rules to the configuration mode route-map:

- **match (route-map configuration mode)** adds a match rule to a route map.
- **set (route-map configuration mode)** adds a set rule to a route map.

Editing a Clause

To append a rule to a list, enter the rule without a sequence number in route map configuration mode for the list. The new rule's sequence number is derived by adding 10 to the last rule's sequence number.

Examples

- These commands enter route map configuration mode for an existing route map clause, then adds a set and match statement to the clause.

```
switch(config)#route-map Map1 permit 20
switch(config-route-map-Map1)#set ip next-hop 10.2.4.5
switch(config-route-map-Map1)#match tag 500
switch(config-route-map-Map1)#
```

This command displays the contents of the clause before saving the statements.

```
switch(config-route-map-Map1)#show
route-map Map1 deny 10
  Match clauses:
    match as 10
    match tag 333
  Set clauses:
    set local-preference 100
route-map Map1 permit 20
  Match clauses:
    match metric-type type-1
    match as-path List1
  Set clauses:
```

This command exits route map configuration mode, saves the new statements, and displays the contents of the clause after the statements are saved.

```
switch(config-route-map-Map1)#exit
switch(config)#show route-map Map1
route-map Map1 deny 10
  Match clauses:
    match as 10
    match tag 333
  Set clauses:
    set local-preference 100
route-map Map1 permit 20
  Match clauses:
    match metric-type type-1
    match as-path List1
    match tag 500
  Set clauses:
    set ip next-hop 10.2.4.5
switch(config)#
```

Inserting a Clause

To insert a new clause into an existing route map, create a new clause with a sequence number that differs from any existing clause in the map.

Example

- This command adds clause 50 to the ***Map1*** route map, then displays the new route map.

```
switch(config)#route-map Map1 permit 50
switch(config-route-map-Map1)#match as 150
switch(config-route-map-Map1)#exit
switch(config)#show route-map Map1
route-map Map1 deny 10
  Match clauses:
    match as 10
    match tag 333
  Set clauses:
    set local-preference 100
route-map Map1 permit 50
  Match clauses:
    match as 150
  Set clauses:
switch(config)#
```

Deleting Route Map Components

To remove a component from a route map, perform one of the following:

- To remove a statement from a clause, enter **no**, followed by the statement to be removed.
- To remove a clause, enter **no** followed by the sequence number of the clause to be removed.
- To remove a route map, enter **no** followed by the route map without a sequence number.

13.3.3 Using Route Maps

Protocol redistribution commands include a route map parameter that determines the routes to be redistributed into the specified protocol domain.

Example

- This command uses ***Map1*** route map to select OSPFv2 routes for redistribution into BGP AS1.

```
switch(config)#router bgp 1
switch(config-router-bgp)#redistribute ospf route-map Map1
switch(config-router-bgp)#exit
switch(config)#
```

13.4 Prefix Lists

A prefix list is an ordered set of rules that defines route redistribution access for a specified IP address space. A prefix list rules consists of a filter action (deny or permit), an address space identifier (IPv4 subnet address or IPv6 prefix), and a sequence number.

Prefix lists are referenced by route-map match commands when filtering routes for redistribution.

- Section 13.4.1 describes the prefix list configuration process.
- Section 13.4.2 describes the application of prefix lists.

13.4.1 Prefix List Configuration

A prefix list is an ordered set of rules that defines route redistribution access for a specified IP address space. A prefix list rule consists of a filter action (deny or permit), a network address (IPv4 subnet or IPv6 prefix), and a sequence number. A rule may also include a alterernate mask size.

The switch supports IPv4 and IPv6 prefix lists. IPv4 lists are constructed in global configuration mode, whereas the switch is placed in a Prefix-list configuration mode to create and edit IPv6 prefix lists.

13.4.1.1 IPv4 Prefix Lists

IPv4 prefix lists are created or modified by adding a IPv4 prefix list rule in global configuration mode. Each rule includes the name of a prefix list, in addition to the sequence number, network address, and filter action. A list consists of all rules that have the same prefix list name.

The **ip prefix-list** command creates a prefix list or adds a rule to an existing list. Route map match statements use prefix lists to filter routes for redistribution into OSPF, RIP, or BGP domains.

Example

- These commands creates two IPv4 prefix lists: a two-rule list named route-one and a three-rule list named route-two.

```
switch(config)#ip prefix-list route-one seq 10 deny 10.1.1.1/24
switch(config)#ip prefix-list route-one seq 20 deny 10.1.2.1/16
switch(config)#ip prefix-list route-two seq 10 deny 10.1.1.0/24 ge 26 le 30
switch(config)#ip prefix-list route-two seq 20 deny 10.1.0.0/16
switch(config)#ip prefix-list route-two seq 30 permit 12.15.4.9/32
switch(config)#ip prefix-list route-two seq 40 deny 1.1.1.0/24
switch(config)#show running-config
<-------OUTPUT OMITTED FROM EXAMPLE-------->
!
ip prefix-list route-one seq 10 deny 10.1.1.0/24
ip prefix-list route-one seq 20 deny 10.1.0.0/16
ip prefix-list route-two seq 10 deny 10.1.1.0/24 ge 26 le 30
ip prefix-list route-two seq 20 deny 10.1.0.0/16
ip prefix-list route-two seq 30 permit 12.15.4.9/32
ip prefix-list route-two seq 40 deny 1.1.1.0/24
!
<-------OUTPUT OMITTED FROM EXAMPLE-------->
switch(config)#
```

IPv4 prefix lists are referenced in route-map **match (route-map configuration mode)** commands.

13.4.1.2 IPv6 Prefix Lists

Creating an IPv6 Prefix List

The switch provides IPv6 prefix-list configuration mode for creating and modifying IPv6 prefix lists. A list's can be edited only in the mode where it was created.

To create an IP ACL, enter the **ipv6 prefix-list** command, followed by the name of the list.

The switch enters IPv6 prefix-list configuration mode for the list. If the command is followed by the name of an existing ACL, subsequent commands edit that list.

Example

- This command places the switch in IPv6 prefix list configuration mode to create an IPv6 prefix list named ***map1***.

```
switch(config)#ipv6 prefix-list map1
switch(config-ipv6-pfx)#
```

Adding a Rule

To append a rule to the end of a list, enter the rule without a sequence number while in Prefix-List configuration mode for the list. The new rule's sequence number is derived by adding 10 to the last rule's sequence number.

Examples

- These commands enter the first two rules into a new prefix list.

```
switch(config-ipv6-pfx)#permit 3:4e96:8ca1:33cf::/64
switch(config-ipv6-pfx)#permit 3:11b1:8fe4:1aac::/64
```

To view the list, save the rules by exiting the prefix-list command mode, then re-enter the configuration mode and type **show active**.

```
switch(config-ipv6-pfx)#exit
switch(config)#ipv6 prefix-list map1
switch(config-ipv6-pfx)#show active
ipv6 prefix-list map1
   seq 10 permit 3:4e96:8ca1:33cf::/64
   seq 20 permit 3:11b1:8fe4:1aac::/64
switch(config-ipv6-pfx)#
```

This command appends a rule to the end of the prefix list. The sequence number of the new rule is 30.

```
switch(config-ipv6-pfx)#permit 3:1bca:1141:ab34::/64
switch(config-ipv6-pfx)#exit
switch(config)#ipv6 prefix-list map1
switch(config-ipv6-pfx)#show active
ipv6 prefix-list map1
   seq 10 permit 3:4e96:8ca1:33cf::/64
   seq 20 permit 3:11b1:8fe4:1aac::/64
   seq 30 permit 3:1bca:1141:ab34::/64
switch(config-ipv6-pfx)#
```

Inserting a Rule

To insert a rule into a prefix list, use the **seq (IPv6 Prefix Lists)** command to enter a rule with a sequence number that is between numbers of two existing rules.

Example

- This command inserts a rule between the first two rules by assigning it the sequence number 15.

```
switch(config-ipv6-pfx)#seq 15 deny 3:4400::/64
switch(config-ipv6-pfx)#exit
switch(config)#show ipv6 prefix-list map1
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 20 permit 3:11b1:8fe4:1aac::/64
seq 30 permit 3:1bca:3ff2:634a::/64
switch(config)#
```

Deleting a Rule

To remove a rule from the configuration mode prefix list, enter **no seq** (see **seq (IPv6 Prefix Lists)**), followed by the sequence number of the rule to be removed.

Example

- These commands remove rule 20 from the prefix list, then displays the resultant prefix list.

```
switch(config-ipv6-pfx)#no seq 20
switch(config-ipv6-pfx)#exit
switch(config)#show ipv6 prefix-list map1
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 30 permit 3:1bca:3ff2:634a::/64
switch(config)#
```

13.4.2 Using Prefix Lists

Route map match statement include an option that matches on a specified prefix lists.

Example

- The MAP_1 route map uses a match statement that references the PL_1 prefix list.

```
switch(config)#route-map MAP_1 permit
switch(config-route-map-MAP_1)#match ip address prefix-list PL_1
switch(config-route-map-MAP_1)#set community 500
switch(config-route-map-MAP_1)#exit
```

13.5 ACL, Route Map, and Prefix List Commands

This section describes CLI commands that this chapter references.

Implementation Commands

- ip access-list . . . . . Page 494
- ip access-list standard . . . . . Page 495
- ip prefix-list . . . . . Page 496
- ipv6 access-list . . . . . Page 498
- ipv6 access-list standard . . . . . Page 499
- ipv6 prefix-list . . . . . Page 500
- mac access-list . . . . . Page 502

Interface and Control Plane Configuration Mode Commands

- ip access-group (Interface Configuration modes) . . . . . Page 493
- ipv6 access-group . . . . . Page 497
- mac access-group . . . . . Page 501

ACL Edit Commands

- resequence . . . . . Page 515
- no <sequence number> . . . . . Page 504
- show (ACL configuration modes) . . . . . Page 520
- statistics per-entry (ACL configuration modes) . . . . . Page 529

ACL Rule Commands

- deny (IPv4 ACL) . . . . . Page 484
- deny (IPv6 ACL) . . . . . Page 486
- deny (MAC ACL) . . . . . Page 489
- deny (Standard IPv4 ACL) . . . . . Page 491
- deny (Standard IPv6 ACL) . . . . . Page 492
- permit (IPv4 ACL) . . . . . Page 505
- permit (IPv6 ACL) . . . . . Page 507
- permit (MAC ACL) . . . . . Page 510
- permit (Standard IPv4 ACL) . . . . . Page 512
- permit (Standard IPv6 ACL) . . . . . Page 513
- remark . . . . . Page 514

ACL List Counter Reset Command

- clear ip access-lists counters . . . . . Page 481
- clear ipv6 access-lists counters . . . . . Page 482

IPv6 Prefix List Configuration Commands

- deny (IPv6 Prefix List) . . . . . Page 488
- permit (IPv6 Prefix List) . . . . . Page 509
- seq (IPv6 Prefix Lists) . . . . . Page 518

Route Map Edit Commands

- continue (route-map configuration mode) . . . . . Page 483
- match (route-map configuration mode) . . . . . Page 503
- route-map . . . . . Page 516
- set (route-map configuration mode) . . . . . Page 519
- show (route-map configuration mode) . . . . . Page 522

Display Commands

- show ip access-lists . . . . . Page 523
- show ip prefix-list . . . . . Page 524
- show ipv6 access-lists . . . . . Page 525
- show ipv6 prefix-list . . . . . Page 526
- show mac access-lists . . . . . Page 527
- show route-map . . . . . Page 528

clear ip access-lists counters

The **clear ip access-lists counters** command sets ACL counters to zero for the specified IPv4 access control list (ACL). The ***session*** parameter limits ACL counter clearing to the current CLI session.

Command Mode

Global Configuration

Command Syntax

```
clear ip access-lists counters [ACL_NAME] [SCOPE]
```

Parameters

- ***ACL_NAME*** Name of ACL. Options include:
  - <no parameter> all ACLs.
  - *access_list* name of ACL.
- ***SCOPE*** Session affected by command. Options include:
  - <no parameter> command affects counters on all CLI sessions.
  - **session** affects only current CLI session.

Example

- This command resets all IPv4 ACL counters.

```
switch(config)#clear ip access-lists counters
switch(config)#
```

clear ipv6 access-lists counters

The **clear ipv6 access-lists counters** command sets ACL counters to zero for the specified IPv6 access control list (ACL). The ***session*** parameter limits ACL counter clearing to the current CLI session.

Command Mode

Privileged EXEC

Command Syntax

```
clear ipv6 access-lists counters [ACL_NAME] [SCOPE]
```

Parameters

- *ACL_NAME* name of ACL. Options include:
  - <no parameter> all IPv6 ACLs.
  - *access_list* name of IPv6 ACL.
- ***SCOPE*** Session affected by command. Options include:
  - <no parameter> command affects counters on all CLI sessions.
  - **session** affects only current CLI session.

Example

- This command resets all IPv6 ACL counters.

```
switch(config)#clear ipv6 access-lists counters
switch(config)#
```

continue (route-map configuration mode)

The **continue** command creates a route map clause entry that enables additional route map evaluation of routes whose parameters meet the clause's matching crieteria.

A clause typically contains a **match (route-map configuration mode)** and a **set (route-map configuration mode)** statement. The evaluation of routes whose settings are the same as **match** statement parameters normally end and the clause's **set** statement are applied to the route. Routes that match a clause containing a **continue** statement are evaluated against the clause specified by the continue statement.

When a route matches multiple route-map clauses, the filter action (deny or permit) is determined by the last clause that the route matches. The **set** statements in all clauses matching the route are applied to the route after the route map evaluation is complete. Multiple set statements are applied in the same order by which the route was evaluated against the clauses containing them.

The **no continue** and **default continue** commands remove the correponding **continue** statement from the configuration mode route map clause by deleting the corresponding command from ***running-config***.

Command Mode

Route-MapConfiguration

Command Syntax

```
continue NEXT_SEQ
no continue NEXT_SEQ
default continue NEXT_SEQ
```

Parameters

- *NEXT_SEQ* specifies number of next clause for evaluating matching routes. Options include:
  - <no parameter> Next clause in the route map, as determined by sequence number.
  - *seq_number* Specifies the number of the next clause. Values range from 1 to 16777215.

Restrictions

A continue statement cannot specify a sequence number smaller than the sequence number of its route-map clause.

Example

- This command creates route map map1, clause 40 with a match statement, a set statement, and a continue statement. Routes that match the clause are subsequently evaluated agains clause 100. The **set local-preference** statement is applied to matching routes regardless of subsequent matching operations.

```
switch(config)#route-map map1 deny 40
switch(config-route-map-map1)#match as 15
switch(config-route-map-map1)#continue 100
switch(config-route-map-map1)#set local-preference 50
switch(config-route-map-map1)#
```

deny (IPv4 ACL)

The **deny** command adds a deny rule to the configuration mode IPv4 access control list (ACL). Packets filtered by a **deny** rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The **no <sequence number>** command also removes the specified rule from the ACL.

Command Mode

ACL Configuration

Command Syntax

```
[SEQ_NUM] deny PROTOCOL SOURCE_ADDR [SOURCE_PORT] DEST_ADDR [DEST_PORT]
[fragments] [FLAGS] [MESSAGE] [tracked] [log] [TTL_FILTER]

no deny PROTOCOL SOURCE_ADDR [SOURCE_PORT] DEST_ADDR [DEST_PORT]
[fragments] [FLAGS] [MESSAGE] [tracked] [log] [TTL_FILTER]

default deny PROTOCOL SOURCE_ADDR [SOURCE_PORT] DEST_ADDR [DEST_PORT]
[fragments] [FLAGS] [MESSAGE] [tracked] [log] [TTL_FILTER]
```

Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a deny rule.

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - ***<1 – 4294967295>*** Number assigned to entry.
- ***PROTOCOL*** protocol field filter. Values include:
  - **ahp** authentication header protocol (51).
  - **icmp** internet control message protocol (1).
  - **igmp** internet group management protocol (2).
  - **ip** internet protocol – IPv4 (4).
  - **ospf** open shortest path first (89).
  - **pim** protocol independent multicast (103).
  - **tcp** transmission control protocol (6).
  - **udp** user datagram protocol (17).
  - **vrrp** virtual router redundancy protocol (112).
  - *protocol_num* integer corresponding to an IP protocol. Values range from 0 to 255.
- ***SOURCE_ADDR*** and ***DEST_ADDR*** source and destination address filters. Options include:
  - *network_addr* subnet address (CIDR or address-mask).
  - **any** Packets from all addresses are filtered.
  - **host** *ip_addr* IP address (dotted decimal notation).

  Subnet addresses support discontiguous masks.
- ***SOURCE_PORT*** and ***DEST_PORT*** source and destination port filters. Options include:
  - **any** all ports
  - **eq** *port-1 port-2 ... port-n* A list of ports. Maximum list size is 10 ports.
  - **neq** *port-1 port-2 ... port-n* The set of all ports not listed. Maximum list size is 10 ports.

  - — **gt** *port* The set of ports with larger numbers than the listed port.
  - — **lt** *port* The set of ports with smaller numbers than the listed port.
  - — **range** *port_1 port_2* The set of ports whose numbers are between the range.
- **fragments** filters packets with FO bit set (indicates a non-initial fragment packet).
- ***FLAGS*** flag bit filters (TCP packets).
  - — Use CLI syntax assistance (?) to display available options.
- ***MESSAGE*** message type filters (ICMP packets).
  - — Use CLI syntax assistance (?) to display available options.
- **tracked** rule filters packets in existing ICMP, UDP, or TCP connections.

  Valid in ACLs applied to the control plane.
  Validity in ACLs applied to data plane varies by switch platform.
- **log** triggers an informational log message to the console about the matching packet.

  Valid in ACLs applied to the control plane.
  Validity in ACLs applied to data plane varies by switch platform.
- ***TTL_FILTER*** filters by packet's TTL (time-to-live) value. Values include:
  - — **ttl eq** *ttl_value* Packets match if ***ttl*** in packet is equal to *ttl_value*.
  - — **ttl gt** *ttl_value* Packets match if ***ttl*** in packet is greater than *ttl_value*.
  - — **ttl lt** *ttl_value* Packets match if ***ttl*** in packet is less than *ttl_value*.
  - — **ttl neq** *ttl_value* Packets match if ***ttl*** in packet is not equal to *ttl_value*.

  Valid in ACLs applied to the control plane.
  Validity in ACLs applied to data plane varies by switch platform.

Examples

- This command appends a **deny** statement at the end of the ACL. The **deny** statement drops OSPF packets from 10.10.1.1/24 to any host.

```
switch(config)#ip access-list text1
switch(config-acl-text1)#deny ospf 10.1.1.0/24 any
switch(config-acl-text1)#
```

- This command inserts a **deny** statement with the sequence number 65. The **deny** statement drops all PIM packets.

```
switch(config-acl-text1)#65 deny pim any any
switch(config-acl-text1)#
```

deny (IPv6 ACL)

The **deny** command adds a deny rule to the configuration mode IPv6 access control list (ACL). Packets filtered by a **deny** rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The **no <sequence number>** command also removes the specified rule from the ACL.

Command Mode

IPv6-ACL Configuration

Command Syntax

```
[SEQ_NUM] deny PROT SRC_ADDR [SRC_PT] DEST_ADDR [DEST_PT][HOP][FLAG][MSG]
no deny PROT SRC_ADDR [SRC_PT] DEST_ADDR [DEST_PT][HOP][FLAG][MSG]
default deny PROT SRC_ADDR [SRC_PT] DEST_ADDR [DEST_PT][HOP][FLAG][MSG]
```

Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a deny rule.

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - ***<1 – 4294967295>*** Number assigned to entry.
- ***PROT*** protocol field filter. Values include:
  - **icmpv6** Internet Control Message Protocol for version 6 (58).
  - **ipv6** Internet Protocol – IPv6 (41).
  - **tcp** Transmission Control Protocol (6).
  - **udp** User Datagram Protocol (17).
  - *protocol_num* integer corresponding to an IP protocol. Values range from 0 to 255.
- ***SRC_ADDR*** and ***DEST_ADDR*** source and destination address filters. Options include:
  - *ipv6_prefix* IPv6 address with prefix length (CIDR notation).
  - **any** Packets from all addresses are filtered.
  - **host** *ipv6_addr* IPv6 host address.
- ***SRC_PT*** and ***DEST_PT*** source and destination port filters. Options include:
  - **any** all ports
  - **eq** *port-1 port-2 ... port-n* A list of ports. Maximum list size is 10 ports.
  - **neq** *port-1 port-2 ... port-n* The set of all ports not listed. Maximum list size is 10 ports.
  - **gt** *port* The set of ports with larger numbers than the listed port.
  - **lt** *port* The set of ports with smaller numbers than the listed port.
  - **range** *port_1 port_2* The set of ports whose numbers are between the range.
- ***HOP*** filters by packet's hop-limit value. Options include:
  - **hop-limit eq** *hop_value* Packets match if ***hop-limit*** value in packet equals *ttl_value*.
- ***FLAG*** flag bit filters (TCP packets).
  - Use CLI syntax assistance (?) to display available options.

- *MSG* message type filters (ICMPv6 packets).
  - Use CLI syntax assistance (?) to display available options.

Example

- This command appends a **deny** statement at the end of the ACL. The **deny** statement drops IPv6 packets from 3710:249a:c643:ef11::/64 to any host.

```
switch(config)#ipv6 access-list text1
switch(config-acl-text1)#deny ipv6 3710:249a:c643:ef11::/64 any
switch(config-acl-text1)#
```

deny (IPv6 Prefix List)

The **deny** command adds a rule to the configuration mode IPv6 prefix list. Route map match statements use prefix lists to filter routes for redistribution into OSPF, RIP, or BGP domains. Routes are denied access when they match the prefix that a **deny** statement specifies.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode prefix list. The **no seq (IPv6 Prefix Lists)** command also removes the specified rule from the prefix list.

Command Mode

IPv6-pfx Configuration

Command Syntax

```
[SEQUENCE] deny ipv6_prefix [MASK]
```

Parameters

- *SEQUENCE* Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the list's last rule.
  - **seq** *seq_num* Number is specified by *seq_num*. Value ranges from 0 to 65535.
- *ipv6_prefix* IPv6 prefix upon which command filters routes (CIDR notation).
- *MASK* range of the prefix length to be matched for prefixes that are more specific than the network parameter.
  - <no parameter> exact match with the subnet mask is required.
  - **ge** *mask_g* range is from *mask_g* to 128.
  - **le** *mask_l* range is from *subnet* mask length to *mask_l*.
  - **ge** *mask_l* **le** *mask_g* range is from *mask_g* to *mask_l*.

    mask_l and *mask_g* range from 1 to 128.

    when **le** and **ge** are specified, *subnet* mask > *mask_g*>*mask_l*

Example

- This command appends a **deny** statement at the end of the text1 prefix list. The **deny** statement denies redistribution of routes with the specified prefix.

```
switch(config)#ipv6 prefix-list route-five
switch(config-ipv6-pfx)#deny 3100::/64
switch(config-ipv6-pfx)#
```

deny (MAC ACL)

The **deny** command adds a deny rule to the configuration mode MAC access control list (ACL). Packets filtered by a deny rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The **no <sequence number>** command also removes the specified rule from the ACL.

Command Mode

MAC-ACL Configuration

Command Syntax

```
[SEQ_NUM] deny SOURCE_ADDR DEST_ADDR [PROTOCOL] [log]
no deny SOURCE_ADDR DEST_ADDR [PROTOCOL] [log]
default deny SOURCE_ADDR DEST_ADDR [PROTOCOL] [log]
```

Parameters

- *SEQ_NUM* Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - *<1 – 4294967295>* Number assigned to entry.
- *SOURCE_ADDR* and *DEST_ADDR* source and destination address filters. Options include:
  - *mac_address mac_mask* MAC address and mask
  - **any** Packets from all addresses are filtered.

  mac_address specifies a MAC address in 3x4 dotted hexadecimal notation (hhhh.hhhh.hhhh)
  mac_mask specifies a MAC address mask in 3x4 dotted hexadecimal notation (hhhh.hhhh.hhhh)
  - *0* bits require an exact match to filter
  - *1* bits filter on any value
- *PROTOCOL* protocol field filter. Values include:
  - **aarp** Appletalk Address Resolution Protocol (0x80f3)
  - **appletalk** Appletalk (0x809b)
  - **arp** Address Resolution Protocol (0x806)
  - **ip** Internet Protocol Version 4 (0x800)
  - **ipx** Internet Packet Exchange (0x8137)
  - **lldp** LLDP (0x88cc)
  - **novell** Novell (0x8138)
  - **rarp** Reverse Address Resolution Protocol (0x8035)
  - *protocol_num* integer corresponding to a MAC protocol. Values range from 0 to 65535
- **log** triggers an informational log message to the console about the matching packet.

Examples

- This command appends a permit statement at the end of the ACL. The deny statement drops all aarp packets from 10.1000.0000 through 10.1000.FFFF to any host.

```
switch(config)#mac access-list text1
switch(config-mac-acl-text1)#deny 10.1000.0000 0.0.FFFF any aarp
switch(config-mac-acl-text1)#
```

- This command inserts a permit statement with the sequence number 25. The deny statement drops all packets through the interface.

```
switch(config-mac-acl-text1)#25 deny any any
switch(config-mac-acl-text1)#
```

deny (Standard IPv4 ACL)

The **deny** command adds a deny rule to the configuration mode standard IPv4 access control list (ACL). Standard ACL rules filter on the source field.

Packets filtered by a **deny** rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The **no <sequence number>** command also removes the specified rule from the ACL.

Command Mode

Std-ACL Configuration

Command Syntax

```
[SEQ_NUM] deny SOURCE_ADDR [log]
no deny SOURCE_ADDR [log]
default deny SOURCE_ADDR [log]
```

Parameters

- *SEQ_NUM* Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - *<1 – 4294967295>* Number assigned to entry.
- *SOURCE_ADDR* source address filter. Options include:
  - *network_addr* subnet address (CIDR or address-mask).
  - **any** packets from all addresses are filtered.
  - **host** *ip_addr* IP address (dotted decimal notation).

  Subnet addresses support discontiguous masks.
- **log** triggers an informational log message to the console about the matching packet.

  Valid in ACLs applied to the control plane.
  Validity in ACLs applied to data plane varies by switch platform.

Example

- This command appends a **deny** statement at the end of the ACL. The **deny** statement drops packets from 10.10.1.1/24.

```
switch(config)#ip access-list standard text1
switch(config-std-acl-text1)#deny 10.1.1.1/24
switch(config-std-acl-text1)#
```

deny (Standard IPv6 ACL)

The **deny** command adds a deny rule to the configuration mode standard IPv6 access control list (ACL). Standard ACL rules filter on the source field.

Packets filtered by a **deny** rule are dropped by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no deny** and **default deny** commands remove the specified rule from the configuration mode ACL. The **no <sequence number>** command also removes the specified rule from the ACL.

Command Mode

Std-IPv6-ACL Configuration

Command Syntax

```
[SEQ_NUM] deny SOURCE_ADDR
no deny SOURCE_ADDR
default deny SOURCE_ADDR
```

Parameters

- *SEQ_NUM* Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - *<1 – 4294967295>* Number assigned to entry.
- *SOURCE_ADDR* source address filter. Options include:
  - *ipv6_prefix* IPv6 address with prefix length (CIDR notation).
  - **any** Packets from all addresses are filtered.
  - **host** *ipv6_addr* IPv6 host address.

Example

- This command appends a **deny** statement at the end of the ACL. The **deny** statement drops packets from 2103::/64.

```
switch(config)#ipv6 access-list standard text1
switch(config-std-acl-ipv6-text1)#deny 2103::/64
switch(config-std-acl-ipv6-text1)#
```

ip access-group (Interface Configuration modes)

The **ip access-group** command applies an IPv4 or standard IPv4 access control list (ACL) to the configuration mode interface or control plane.

The **no ip access-group** and **default ip access-group** commands remove the corresponding **ip access-group** command from ***running-config***.

Command Mode

Interface Ethernet Configuration
Interface Port Channel Configuration
Interface VLAN Configuration (Trident platform only)

Command Syntax

```
ip access-group list_name DIRECTION
no ip access-group list_name DIRECTION
default ip access-group list_name DIRECTION
```

Parameters

- *list_name* name of ACL assigned to interface.
- ***DIRECTION*** transmission direction of packets, relative to interface. Valid options include:
  - **in** inbound packets.
  - **out** outbound packets.

Example

- These commands applies the IPv4 ACL named ***test2*** to the Ethernet 3 interface.

```
switch(config)#interface ethernet 3
switch(config-if-Et3)#ip access-group test2 in
switch(config-if-Et3)#
```

ip access-list

The **ip access-list** command places the switch in ACL configuration mode, which is a group change mode that modifies an IPv4 access control list. The command specifies the name of the IPv4 ACL that subsequent commands modify and creates an ACL if it references a nonexistent list. All changes in a group change mode edit session are pending until the end of the session.

The **exit** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **abort** command discards pending ACL changes, returning the switch to global configuration mode.

The **no ip access-list** and **default ip access-list** commands delete the specified IPv4 ACL.

Command Mode

Global Configuration

Command Syntax

```
ip access-list list_name
no ip access-list list_name
default ip access-list list_name
```

Parameters

- *list_name* Name of ACL.
  Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in ACL configuration mode:

- **deny (IPv4 ACL)**
- **no <sequence number>**
- **permit (IPv4 ACL)**
- **remark**
- **resequence**
- **show (ACL configuration modes)**

Related Commands

- **ip access-list standard** enters std-acl configuration mode for editing standard IP ACLs.
- **show ip access-lists** displays IP and standard ACLs.

Examples

- This command places the switch in ACL configuration mode to modify the ***filter1*** IPv4 ACL.

```
switch(config)#ip access-list filter1
switch(config-acl-filter1)#
```

- This command saves changes to ***filter1*** ACL, then returns the switch to global configuration mode.

```
switch(config-acl-filter1)#exit
switch(config)#
```

- This command discards changes to ***filter1***, then returns the switch to global configuration mode.

```
switch(config-acl-filter1)#abort
switch(config)#
```

ip access-list standard

The **ip access-list standard** command places the switch in std-ACL configuration mode, which is a group change mode that modifies a standard IPv4 access control list. The command specifies the name of the standard IPv4 ACL that subsequent commands modify, and creates an ACL if it references a nonexistent list. All group change mode edit session changes are pending until the session ends.

The **exit** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. Pending changes are also saved by entering a different configuration mode.

The **abort** command discards pending ACL changes, returning the switch to global configuration mode.

The **no ip access-list standard** and **default ip access-list standard** commands delete the specified ACL.

Command Mode

Global Configuration

Command Syntax

```
ip access-list standard list_name
no ip access-list standard list_name
default ip access-list standard list_name
```

Parameters

- *list_name* Name of standard ACL.
  Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in std-ACL configuration mode:

- **deny (Standard IPv4 ACL)**
- **no <sequence number>**
- **permit (Standard IPv4 ACL)**
- **remark**
- **resequence**
- **show (ACL configuration modes)**

Related Commands

- **ip access-list** enters ACL configuration mode for editing IPv4 ACLs.
- **show ip access-lists** displays IPv4 and standard IPv4 ACLs.

Examples

- This command places the switch in std-ACL configuration mode to modify the ***filter2*** IPv4 ACL.

```
switch(config)#ip access-list standard filter2
switch(config-std-acl-filter2)#
```

- This command saves changes to ***filter2*** ACL, then returns the switch to global configuration mode.

```
switch(config-std-acl-filter2)#exit
switch(config)#
```

- This command discards changes to ***filter2***, then returns the switch to global configuration mode.

```
switch(config-std-acl-filter2)#abort
switch(config)#
```

ip prefix-list

The **ip prefix-list** command creates a prefix list or adds an entry to an existing list. Route map match statements use prefix lists to filter routes for redistribution into OSPF, RIP, or BGP domains.

A prefix list comprises all prefix list entries with the same label. The sequence numbers of the rules in a prefix list specify the order that the rules are applied to a route that the match statement is evaluating.

The **no ip prefix-list** and **default ip prefix-list** commands delete the specified prefix list entry by removing the corresponding **ip prefix-list** statement from ***running-config***. If the **no** or **default ip prefix-list** command does not list a sequence number, the command deletes all entries of the prefix list.

Command Mode

Global Configuration

Command Syntax

```
ip prefix-list list_name [SEQUENCE] FILTER_TYPE network_addr [MASK]
no ip prefix-list list_name [SEQUENCE]
default ip prefix-list list_name [SEQUENCE]
```

Parameters

- *list_name* The label that identifies the prefix list.
- ***SEQUENCE*** Sequence number of the prefix list entry. Options include
  - — <no parameter> entry's number is ten plus highest sequence number in current list.
  - — **seq** *seq_num* number assigned to entry. Value ranges from 0 to 65535.
- ***FILTER_TYPE*** specifies route access when it matches IP prefix list. Options include:
  - — **permit** routes are permitted access when they match the specified subnet.
  - — **deny** routes are denied access when they match the specified subnet.
- *network_addr* Subnet upon which command filters routes. Format is CIDR or address-mask.
- ***MASK*** range of the prefix length to be matched for prefixes that are more specific than the network parameter.
  - — <no parameter> exact match with the subnet mask is required.
  - — **ge** *mask_g* range is from *mask_g* to 32.
  - — **le** *mask_l* range is from *subnet* mask length to *mask_l*.
  - — **ge** *mask_l* **le** *mask_g* range is from *mask_g* to *mask_l*.

    mask_l and *mask_g* range from 1 to 32.

    when **le** and **ge** are specified, *subnet* mask > *mask_g*>*mask_l*

Example

- These commands create a two-entry prefix list named route-one.

```
switch(config)#ip prefix-list route-one seq 10 deny 10.1.1.1/24 ge 26 le 30
switch(config)#ip prefix-list route-one seq 20 deny 10.1.2.1/16
switch(config)#
```

ipv6 access-group

The **ipv6 access-group** command applies an IPv6 or standard IPv6 access control list (ACL) to the configuration mode interface.

The **no ipv6 access-group** and **default ipv6 access-group** commands remove the corresponding **ipv6 access-group** command from ***running-config***.

Command Mode

Interface Ethernet Configuration
Interface Port Channel Configuration
Interface VLAN Configuration

Command Syntax

```
ipv6 access-group list_name DIRECTION
no ipv6 access-group list_name DIRECTION
default ipv6 access-group list_name DIRECTION
```

Parameters

- *list_name* name of ACL assigned to interface.
- ***DIRECTION*** transmission direction of packets, relative to interface. Valid options include:
  - **in** inbound packets.

Examples

- These commands assign the IPv6 ACL named ***test2*** to the Ethernet 3 interface.

```
switch(config)#interface ethernet 3
switch(config-if-Et3)#ipv6 access-group test2 in
switch(config-if-Et3)#
```

ipv6 access-list

The **ipv6 access-list** command places the switch in IPv6-ACL configuration mode, which is a group change mode that modifies an IPv6 access control list. The command specifies the name of the IPv6 ACL that subsequent commands modify and creates an ACL if it references a nonexistent list. All changes in a group change mode edit session are pending until the end of the session.

The **exit** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **abort** command discards pending ACL changes, returning the switch to global configuration mode.

The **no ipv6 access-list** and **default ipv6 access-list** commands delete the specified IPv6 ACL.

Command Mode

Global Configuration

Command Syntax

```
ipv6 access-list list_name
no ipv6 access-list list_name
default ipv6 access-list list_name
```

Parameters

- *list_name* Name of ACL.
  Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in IPv6-ACL configuration mode:

- **deny (IPv6 ACL)**
- **no <sequence number>**
- **permit (IPv6 ACL)**
- **remark**
- **resequence**
- **show (ACL configuration modes)**

Related Commands

- **ipv6 access-list standard** enters std-ipv6-acl configuration mode for editing standard IPv6 ACLs.
- **show ipv6 access-lists** displays IPv6 and standard IPv6 ACLs.

Examples

- This command places the switch in IPv6-ACL configuration mode to modify the ***filter1*** IPv6 ACL.

```
switch(config)#ipv6 access-list filter1
switch(config-ipv6-acl-filter1)#
```

- This command saves changes to ***filter1*** ACL, then returns the switch to global configuration mode.

```
switch(config-ipv6-acl-filter1)#exit
switch(config)#
```

- This command discards changes to ***filter1***, then returns the switch to global configuration mode.

```
switch(config-ipv6-acl-filter1)#abort
switch(config)#
```

ipv6 access-list standard

The **ipv6 access-list standard** command places the switch in std-IPv6-ACL-configuration mode, which is a group change mode that modifies a standard IPv6 access control list. The command specifies the name of the standard IPv6 ACL that subsequent commands modify and creates an ACL if it references a nonexistent list. All group change mode edit session changes are pending until the session ends.

The **exit** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. Pending changes are also saved by entering a different configuration mode.

The **abort** command discards pending ACL changes, returning the switch to global configuration mode.

The **no ipv6 access-list standard** and **default ipv6 access-list standard** commands delete the specified ACL.

Command Mode

Global Configuration

Command Syntax

```
ipv6 access-list standard list_name
no ipv6 access-list standard list_name
default ipv6 access-list standard list_name
```

Parameters

- *list_name* Name of ACL.
  Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in std-IPv6-ACL configuration mode:

- **deny (Standard IPv6 ACL)**
- **no <sequence number>**
- **permit (Standard IPv6 ACL)**
- **remark**
- **resequence**
- **show (ACL configuration modes)**

Related Commands

- **ipv6 access-list** enters IPv6-ACL configuration mode for editing IPv6 ACLs.
- **show ipv6 access-lists** displays IPv6 and standard IPv6 ACLs.

Examples

- This command places the switch in Std-IPv6 ACL configuration mode to modify the ***filter2*** ACL.

```
switch(config)#ipv6 access-list standard filter2
switch(config-std-ipv6-acl-filter2)#
```

- This command saves changes to ***filter2*** ACL, then returns the switch to global configuration mode.

```
switch(config-std-ipv6-acl-filter2)#exit
switch(config)#
```

- This command discards changes to ***filter2***, then returns the switch to global configuration mode.

```
switch(config-std-ipv6-acl-filter2)#abort
switch(config)#
```

ipv6 prefix-list

The **ip prefix-list** command places the switch in IPv6 prefix-list configuration mode, which is a group change mode that modifies an IPv6 prefix list. The command specifies the name of the IPv6 prefix list that subsequent commands modify and creates a prefix list if it references a nonexistent list. All changes in a group change mode edit session are pending until the end of the session.

The **exit** command saves pending prefix list changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no ipv6 prefix-list** and **default ipv6 prefix-list** commands delete the specified IPv6 prefix list.

Command Mode

Global Configuration

Command Syntax

```
ipv6 prefix-list list_name
no ipv6 prefix-list list_name
default ipv6 prefix-list list_name
```

Parameters

- *list_name* Name of prefix list.
  Must begin with an alphabetic character. Cannot contain spaces or quotation marks.

Commands Available in IPv6-pfx configuration mode:

- **deny (IPv6 Prefix List)**
- **seq (IPv6 Prefix Lists)**
- **permit (IPv6 Prefix List)**

Examples

- This command places the switch in IPv6 prefix-list configuration mode to modify the ***route-five*** prefix list.

```
switch(config)#ipv6 prefix-list route-five
switch(config-ipv6-pfx)#
```

- This command saves changes to the prefix list, then returns the switch to global configuration mode.

```
switch(config-ipv6-pfx)#exit
switch(config)#
```

- This command saves changes to the prefix list, then places the switch in interface-Ethernet mode.

```
switch(config-ipv6-pfx)#interface ethernet 3
switch(config-if-Et3)#
```

- This command discards changes to the prefix list, then returns the switch to global configuration mode.

```
switch(config-ipv6-pfx)#abort
switch(config)#
```

mac access-group

The **mac access-group** command applies a MAC-ACL (access control list) to the configuration mode interface.

The **no mac access-group** and **default mac access-group** commands remove the specified **mac access-group** command from ***running-config***.

Command Mode

Interface Ethernet Configuration
Interface Port Channel Configuration

Command Syntax

```
mac access-group list_name DIRECTION
no mac access-group list_name DIRECTION
default mac access-group list_name DIRECTION
```

Parameters

- *list_name* name of MAC ACL.
- ***DIRECTION*** transmission direction of packets, relative to interface. Valid options include:
  - **in** inbound packets.
  - **out** outbound packets.

Example

- These commands assign the MAC ACL named ***mtest2*** to the Ethernet 3 interface to filter inbound packets.

```
switch(config)#interface ethernet 3
switch(config-if-Et3)#mac access-group mtest2 in
switch(config-if-Et3)#
```

mac access-list

The **mac access-list** command places the switch in MAC-ACL Configuration mode, which is a group change mode that modifies a MAC access control list. The command specifies the name of the MAC ACL that subsequent commands modify and creates an ACL if it references a nonexistent list. All changes in a group change mode edit session are pending until the end of the session.

The **exit** command saves pending ACL changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **abort** command discards pending ACL changes, returning the switch to global configuration mode.

The **no mac access-list** and **default mac access-list** commands delete the specified list.

Command Mode

Global Configuration

Command Syntax

```
mac access-list list_name
no mac access-list list_name
default mac access-list list_name
```

Parameters

- *list_name* Name of MAC ACL.
  Names must begin with an alphabetic character and cannot contain a space or quotation mark.

Commands Available in MAC-ACL configuration mode:

- **deny (MAC ACL)**
- **no <sequence number>**
- **permit (MAC ACL)**
- **remark**
- **resequence**
- **show (ACL configuration modes)**

Examples

- This command places the switch in MAC-ACL configuration mode to modify the ***mfilter1*** MAC ACL.

```
switch(config)#mac access-list mfilter1
switch(config-mac-acl-mfilter1)#
```

- This command saves changes to ***mfilter1*** ACL, then returns the switch to global configuration mode.

```
switch(config-mac-acl-mfilter1)#exit
switch(config)#
```

- This command saves changes to ***mfilter1*** ACL, then places the switch in interface-Ethernet mode.

```
switch(config-mac-acl-mfilter1)#interface ethernet 3
switch(config-if-Et3)#
```

- This command discards changes to ***mfilter1***, then returns the switch to global configuration mode.

```
switch(config-mac-acl-mfilter1)#abort
switch(config)#
```

match (route-map configuration mode)

The **match** command creates a route map clause entry that specifies one condition for evaluating a route. When a clause contains multiple match commands, the permit or deny filter is applied to a route only if its properties are equal to the corresponding parameters in each **match** statement. When a route's properties do not equal the statement parameters, the route is evaluated against the next clause in the route map, as determined by sequence number. If all clauses fail to permit or deny the route, the route is denied.

The **no match** and **default match** commands remove the **match** statement from the configuration mode route map clause by deleting the corresponding command from ***running-config***.

Command Mode

Route-Map Configuration

Command Syntax

```
match CONDITION
no match CONDITION
default match CONDITION
```

Parameters

- *CONDITION* specifies criteria for evaluating a route. Options include:
  - **as** *area_number* BGP autonomous system (1-65535)
  - **as-path** *path_name* BGP autonomous system path access list.
  - **community** *listname* BGP community.
  - **community** *listname* **exact-match** BGP community; list must match set that is present.
  - **extcommunity** *listname* BGP extended community.
  - **extcommunity** *listname* **exact-match** BGP ext. community; list must match set that is present.
  - **interface ethernet** *e_num* specified Ethernet interface.
  - **interface loopback** *l_num* specified loopback interface.
  - **ip address access-list** *acl_name* IPv4 address filtered by Access Control List (ACL).
  - **ip address prefix-list** *plv4_name* IPv4 address filtered by IP prefix list.
  - **ip next-hop** *ipv4_address* next hop IPv4 address.
  - **ipv6 address prefix-list** *plv6_name* IPv6 address filtered by IPv6 prefix list.
  - **ipv6 next-hop** *ipv6_address* next ipv6 hop address.
  - **local-preference** *preference_number* BGP local preference metric (0-4294967295).
  - **metric** *metric_number* route metric (0-4294967295).
  - **metric metric-type type-1** OSPF type 1 metric.
  - **metric metric-type type-2** OSPF type 2 metric.
  - **tag** *tag_number* route tag (0-4294967295).

Example

- This command creates a route-map match rule that filters routes from BGP AS 15.

```
switch(config)#route-map map1
switch(config-route-map-map1)#match as 15
switch(config-route-map-map1)#
```

no <sequence number>

The **no <sequence number>** command removes the rule with the specified sequence number from the ACL. The **default <sequence number>** command also removes the specified rule.

Command Mode

ACL Configuration
IPv6-ACL Configuration
Std-ACL Configuration
Std-IPv6-ACL Configuration
MAC-ACL Configuration

Command Syntax

no *line_num*
default *line_num*

Parameters

- *line_num* sequence number of rule to be deleted. Values range from ***1 to 4294967295***.

Example

- This command removes statement 30 from the list

```
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
switch(config-acl-test1)#no 30
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        40 permit ip any any
        50 remark end of list
```

permit (IPv4 ACL)

The **permit** command adds a permit rule to the configuration mode IPv4 access control list (ACL). Packets filtered by a permit rule are accepted by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The **no <sequence number>** command also removes a specified rule from the ACL.

Command Mode

ACL Configuration

Command Syntax

```
[SEQ_NUM] permit PROTOCOL SOURCE_ADDR [SOURCE_PORT] DEST_ADDR [DEST_PORT]
[fragments] [FLAGS] [MESSAGE] [tracked] [log] [TTL_FILTER]

no permit PROTOCOL SOURCE_ADDR [SOURCE_PORT] DEST_ADDR [DEST_PORT]
[fragments] [FLAGS] [MESSAGE] [tracked] [log] [TTL_FILTER]

default permit PROTOCOL SOURCE_ADDR [SOURCE_PORT] DEST_ADDR [DEST_PORT]
[fragments] [FLAGS] [MESSAGE] [tracked] [log] [TTL_FILTER]
```

Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a permit rule.

Parameters

- *SEQ_NUM* Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - *<1 – 4294967295>* Number assigned to entry.
- *PROTOCOL* protocol field filter. Values include:
  - **ahp** authentication header protocol (51).
  - **icmp** internet control message protocol (1).
  - **igmp** internet group management protocol (2).
  - **ip** internet protocol – IPv4 (4).
  - **ospf** open shortest path first (89).
  - **pim** protocol independent multicast (103).
  - **tcp** transmission control protocol (6).
  - **udp** user datagram protocol (17).
  - **vrrp** virtual router redundancy protocol (112).
  - *protocol_num* integer corresponding to an IP protocol. Values range from 0 to 255.
- *SOURCE_ADDR* and *DEST_ADDR* source and destination address filters. Options include:
  - *network_addr* subnet address (CIDR or address-mask).
  - **any** Packets from all addresses are filtered.
  - **host** *ip_addr* IP address (dotted decimal notation).

  Source and destination subnet addresses support discontiguous masks.
- *SOURCE_PORT* and *DEST_PORT* source and destination port filters. Options include:
  - **any** all ports
  - **eq** *port-1 port-2 ... port-n* A list of ports. Maximum list size is 10 ports.
  - **neq** *port-1 port-2 ... port-n* The set of all ports not listed. Maximum list size is 10 ports.

  - **gt** *port* The set of ports with larger numbers than the listed port.
  - **lt** *port* The set of ports with smaller numbers than the listed port.
  - **range** *port_1 port_2* The set of ports whose numbers are between the range.
- **fragments** filters packets with FO bit set (indicates a non-initial fragment packet).
- ***FLAGS*** flag bit filters (TCP packets).
  - Use CLI syntax assistance (?) to display available options.
- ***MESSAGE*** message type filters (ICMP packets).
  - Use CLI syntax assistance (?) to display available options.
- **tracked** rule filters packets in existing ICMP, UDP, or TCP connections.

  Valid in ACLs applied to the control plane.
  Validity in ACLs applied to data plane varies by switch platform.
- **log** triggers an informational log message to the console about the matching packet.

  Valid in ACLs applied to the control plane.
  Validity in ACLs applied to data plane varies by switch platform.
- ***TTL_FILTER*** filters by packet's TTL (time-to-live) value. Values include:
  - **ttl eq** *ttl_value* Packets match if ***ttl*** in packet is equal to *ttl_value*.
  - **ttl gt** *ttl_value* Packets match if ***ttl*** in packet is greater than *ttl_value*.
  - **ttl lt** *ttl_value* Packets match if ***ttl*** in packet is less than *ttl_value*.
  - **ttl neq** *ttl_value* Packets match if ***ttl*** in packet is not equal to *ttl_value*.

  Valid in ACLs applied to the control plane.
  Validity in ACLs applied to data plane varies by switch platform.

Examples

- This command appends a **permit** statement at the end of the ACL. The **permit** statement passes all OSPF packets from 10.10.1.1/24 to any host.

```
switch(config)#ip access-list text1
switch(config-acl-text1)#permit ospf 10.1.1.0/24 any
switch(config-acl-text1)#
```

- This command inserts a **permit** statement with the sequence number 25. The **permit** statement passes all PIM packets through the interface.

```
switch(config-acl-text1)#25 permit pim any any
switch(config-acl-text1)#
```

permit (IPv6 ACL)

The **permit** command adds a permit rule to the configuration mode IPv6 access control list (ACL). Packets filtered by a permit rule are accepted by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The **no <sequence number>** command also removes a specified rule from the ACL.

Command Mode

IPv6-ACL Configuration

Command Syntax

```
[SEQ_NUM] permit PROT SRC_ADDR [SRC_PT] DEST_ADDR [DEST_PT][HOP][FLAG][MSG]
no permit PROT SRC_ADDR [SRC_PT] DEST_ADDR [DEST_PT][HOP][FLAG][MSG]
default permit PROT SRC_ADDR [SRC_PT] DEST_ADDR [DEST_PT][HOP][FLAG][MSG]
```

Commands use a subset of the listed fields. Available parameters depend on specified protocol. Use CLI syntax assistance to view options for specific protocols when creating a permit rule.

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - ***<1 – 4294967295>*** Number assigned to entry.
- ***PROT*** protocol field filter. Values include:
  - **icmpv6** Internet Control Message Protocol for v6 (58).
  - **ipv6** Internet Protocol – IPv6 (41).
  - **tcp** Transmission Control Protocol (6).
  - **udp** User Datagram Protocol (17).
  - *protocol_num* integer corresponding to an IP protocol. Values range from 0 to 255.
- ***SRC_ADDR*** and ***DEST_ADDR*** source and destination address filters. Options include:
  - *ipv6_prefix* IPv6 address with prefix length (CIDR notation).
  - **any** Packets from all addresses are filtered.
  - **host** *ipv6_addr* IPv6 host address.
- ***SRC_PT*** and ***DEST_PT*** source and destination port filters. Options include:
  - **any** all ports.
  - **eq** *port-1 port-2 ... port-n* A list of ports. Maximum list size is 10 ports.
  - **neq** *port-1 port-2 ... port-n* The set of all ports not listed. Maximum list size is 10 ports.
  - **gt** *port* The set of ports with larger numbers than the listed port.
  - **lt** *port* The set of ports with smaller numbers than the listed port.
  - **range** *port_1 port_2* The set of ports whose numbers are between the range.
- ***HOP*** filters by packet's hop-limit value. Options include:
  - **hop-limit eq** *hop_value* Packets match if ***hop-limit*** value in packet equals *ttl_value*.
- ***FLAG*** flag bit filters (TCP packets).
  - Use CLI syntax assistance (?) to display available options.

- *MSG* message type filters (ICMPv6 packets).
  - Use CLI syntax assistance (?) to display available options.

Example

- This command appends a **permit** statement at the end of the ACL. The **permit** statement passes all IPv6 packets with the source address 3710:249a:c643:ef11::/64 and with any destination address.

```
switch(config)#ipv6 access-list text1
switch(config-acl-text1)#permit ipv6 3710:249a:c643:ef11::/64 any
switch(config-acl-text1)#
```

permit (IPv6 Prefix List)

The **permit** command adds a rule to the configuration mode IPv6 prefix list. Route map match statements use prefix lists to filter routes for redistribution into OSPF, RIP, or BGP domains. Routes are redistributed into the specified domain when they match the prefix that a **permit** statement specifies.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode prefix list. The **no seq (IPv6 Prefix Lists)** command also removes the specified rule from the prefix list.

Command Mode

IPv6-pfx Configuration

Command Syntax

[***SEQUENCE***] **deny** *ipv6_prefix* [***MASK***]

Parameters

- ***SEQUENCE*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the list's last rule.
  - **seq** *seq_num* Number is specified by *seq_num*. Value ranges from 0 to 65535.
- *ipv6_prefix* IPv6 prefix upon which command filters routes (CIDR notation).
- ***MASK*** range of the prefix length to be matched for prefixes that are more specific than the network parameter.
  - <no parameter> exact match with the subnet mask is required.
  - **ge** *mask_g* range is from *mask_g* to 128.
  - **le** *mask_l* range is from *subnet* mask length to *mask_l*.
  - **ge** *mask_l* **le** *mask_g* range is from *mask_g* to *mask_l*.

    mask_l and *mask_g* range from 1 to 128.

    when **le** and **ge** are specified, the prefix list size > *mask_g*>*mask_l*

Example

- This command appends a **permit** statement at the end of the text1 prefix list. The **permit** statement allows redistribution of routes with the specified prefix.

```
switch(config)#ipv6 prefix-list route-five
switch(config-ipv6-pfx)#permit 3100::/64
switch(config-ipv6-pfx)#
```

permit (MAC ACL)

The **permit** command adds a permit rule to the configuration mode MAC access control list packets through the interface to which the list is applied. Rule filters include protocol, source, and destination.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The **no <sequence number>** command also removes the specified rule from the ACL.

Command Mode

MAC-ACL Configuration

Command Syntax

```
[SEQ_NUM] permit SOURCE_ADDR DEST_ADDR [PROTOCOL] [log]
no permit SOURCE_ADDR DEST_ADDR [PROTOCOL] [log]
default permit SOURCE_ADDR DEST_ADDR [PROTOCOL] [log]
```

Parameters

- ***SEQ_NUM*** Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - ***<1 – 4294967295>*** Number assigned to entry.
- ***SOURCE_ADDR*** and ***DEST_ADDR*** source and destination address filters. Options include:
  - *mac_address mac_mask* MAC address and mask
  - **any** Packets from all addresses are filtered.

    mac_address specifies a MAC address in 3x4 dotted hexadecimal notation (hhhh.hhhh.hhhh)
    mac_mask specifies a MAC address mask in 3x4 dotted hexadecimal notation (hhhh.hhhh.hhhh)
    - *0* bits require an exact match to filter
    - *1* bits filter on any value
- ***PROTOCOL*** protocol field filter. Values include:
  - **aarp** Appletalk Address Resolution Protocol (0x80f3)
  - **appletalk** Appletalk (0x809b)
  - **arp** Address Resolution Protocol (0x806)
  - **ip** Internet Protocol Version 4 (0x800)
  - **ipx** Internet Packet Exchange (0x8137)
  - **lldp** LLDP (0x88cc)
  - **novell** Novell (0x8138)
  - **rarp** Reverse Address Resolution Protocol (0x8035)
  - *protocol_num* integer corresponding to a MAC protocol. Values range from 0 to 65535
- **log** triggers an informational log message to the console about the matching packet.

Examples

- This command appends a **permit** statement at the end of the ACL. The **permit** statement passes all aarp packets from 10.1000.0000 through 10.1000.FFFF to any host.

```
switch(config)#mac access-list text1
switch(config-mac-acl-text1)#permit 10.1000.0000 0.0.FFFF any aarp
switch(config-mac-acl-text1)#
```

- This command inserts a **permit** statement with the sequence number 25. The **permit** statement passes all packets through the interface.

```
switch(config-mac-acl-text1)#25 permit any any
switch(config-mac-acl-text1)#
```

permit (Standard IPv4 ACL)

The **permit** command adds a permit rule to the configuration mode standard IPv4 access control list (ACL). Standard ACL rules filter on the source field.

Packets filtered by a permit rule are accepted by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The **no <sequence number>** command also removes the specified rule from the ACL.

Command Mode

Standard-ACL Configuration

Command Syntax

```
[SEQ_NUM] permit SOURCE_ADDR [log]
no permit SOURCE_ADDR [log]
default permit SOURCE_ADDR [log]
```

Parameters

- *SEQ_NUM* Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - *<1 – 4294967295>* Number assigned to entry.
- *SOURCE_ADDR* source address filter. Options include:
  - *network_addr* subnet address (CIDR or address-mask).
  - **any** Packets from all addresses are filtered.
  - **host** *ip_addr* IP address (dotted decimal notation).

  Subnet addresses support discontiguous masks.
- **log** triggers an informational log message to the console about the matching packet.

Valid in ACLs applied to the control plane.
Validity in ACLs applied to data plane varies by switch platform.

Example

- This command appends a **permit** statement at the end of the ACL. The **permit** statement passes all packets with a source address of 10.10.1.1/24.

```
switch(config)#ip access-list standard text1
switch(config-std-acl-text1)#permit 10.1.1.1/24
switch(config-std-acl-text1)#
```

permit (Standard IPv6 ACL)

The **permit** command adds a permit rule to the configuration mode standard IPv6 access control list. Standard ACL rules filter on the source field.

Packets filtered by a permit rule are accepted by interfaces to which the ACL is applied. Sequence numbers determine rule placement in the ACL. Sequence numbers for commands without numbers are derived by adding 10 to the number of the ACL's last rule.

The **no permit** and **default permit** commands remove the specified rule from the configuration mode ACL. The **no <sequence number>** command also removes the specified rule from the ACL.

Command Mode

Std-IPv6-ACL Configuration

Command Syntax

```
[SEQ_NUM] permit SOURCE_ADDR
no permit SOURCE_ADDR
default permit SOURCE_ADDR
```

Parameters

- *SEQ_NUM* Sequence number assigned to the rule. Options include:
  - <no parameter> Number is derived by adding 10 to the number of the ACL's last rule.
  - *<1 – 4294967295>* Number assigned to entry.
- *SOURCE_ADDR* source address filter. Options include:
  - *ipv6_prefix* IPv6 address with prefix length (CIDR notation).
  - **any** Packets from all addresses are filtered.
  - **host** *ipv6_addr* IPv6 host address.

Example

- This command appends a **permit** statement at the end of the ACL. The **permit** statement drops packets with a source address of 2103::/64.

```
switch(config)#ipv6 access-list standard text1
switch(config-std-acl-ipv6-text1)#permit 2103::/64
switch(config-std-acl-ipv6-text1)#
```

remark

The **remark** command adds a non-executable comment statement into the pending ACL. Remarks entered without a sequence number are appended to the end of the list. Remarks entered with a sequence number are inserted into the list as specified by the sequence number.

The **default remark** command removes the comment statement from the ACL.

The **no remark** command removes the comment statement from the ACL. The command can specify the remark by content or by sequence number.

Command Mode

ACL Configuration
IPv6-ACL Configuration
Std-ACL Configuration
Std-IPv6-ACL Configuration
MAC-ACL Configuration

Command Syntax

```
remark text
line_num remark [text]
no remark text
default remark text
```

Parameters

- *text* the comment text.
- *line_num* sequence number assigned to the remark statement.

Example

- This command appends a comment to the list

```
switch(config-acl-test1)#remark end of list
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
```

resequence

The **resequence** command assigns sequence numbers to rules in the active ACL. Command parameters specify the number of the first rule and the numeric interval between consecutive rules.

Maximum rule sequence number is 4294967295.

Command Mode

ACL Configuration
IPv6-ACL Configuration
Std-ACL Configuration
Std-IPv6-ACL Configuration
MAC-ACL Configuration

Command Syntax

resequence [*start_num* [*inc_num*]]

Parameters

- *start_num* sequence number assigned to the first rule. Default is 10.
- *inc_num* numeric interval between consecutive rules. Default is 10.

Example

- The **resequence** command renumbers the list, starting the first command at number 100 and incrementing subsequent lines by 20.

```
switch(config-acl-test1)#show
IP Access List test1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
switch(config-acl-test1)#resequence 100 20
switch(config-acl-test1)#show
IP Access List test1
        100 permit ip 10.10.10.0/24 any
        120 permit ip any host 10.20.10.1
        140 deny ip host 10.10.10.1 host 10.20.10.1
        160 permit ip any any
        180 remark end of list
```

route-map

The **route-map** command places the switch in route-map configuration mode, which is a group change mode that modifies a route map clause. The command specifies the name and number of the route map clause that subsequent commands modify and creates a route map clause if it references a nonexistent clause. All changes in a group change mode edit session are pending until the end of the session.

Route maps define conditions for redistributing routes between routing protocols. A route map clause is identified by a name, filter type (permit or deny) and sequence number. Clauses with the same name are components of a single route map; the sequence number determines the order in which the clauses are compared to a route.

The **exit** command saves pending route map clause changes to ***running-config***, then returns the switch to global configuration mode. ACL changes are also saved by entering a different configuration mode.

The **abort** command discards pending changes, returning the switch to global configuration mode.

The **no route-map** and **default route-map** commands delete the specified route map clause from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
route-map map_name [FILTER_TYPE] [sequence_number]
no route-map map_name [FILTER_TYPE] [sequence_number]
default route-map map_name [FILTER_TYPE] [sequence_number]
```

Parameters

- *map_name* label assigned to route map. Protocols reference this label to access the route map.
- ***FILTER_TYPE*** disposition of routes matching conditions specified by route map clause.
  - **permit** routes are redistributed when they match route map clause.
  - **deny** routes are not redistributed when they match route map clause.
  - <No parameter> assigns **permit** as the ***FILTER_TYPE***.

  When a route does not match the route map criteria, the next clause within the route map is evaluated to determine the redistribution action for the route.
- *sequence_number* the route map position relative to other clauses with the same name.
  - <no parameter> sequence number of 10 (default) is assigned to the route map.
  - <1-16777215> specifies sequence number assigned to route map.

Commands Available in route-map configuration mode:

- **continue (route-map configuration mode)**
- **match (route-map configuration mode)**
- **set (route-map configuration mode)**

Examples

- This command creates the route map named ***map1*** and places the switch in route-map configuration mode. The route map is configured as a permit map.

```
switch(config)#route-map map1 permit 20
switch(config-route-map-map1)#
```

- This command saves changes to ***map1*** route map, then returns the switch to global configuration mode.

```
switch(config-route-map-map1)#exit
switch(config)#
```

- This command saves changes to ***map1*** route map, then places the switch in interface-Ethernet mode.

```
switch(config-route-map-map1)#interface ethernet 3
switch(config-if-Et3)#
```

- This command discards changes to ***map1*** route map, then returns the switch to global configuration mode.

```
switch(config-route-map-map1)#abort
switch(config)#
```

seq (IPv6 Prefix Lists)

The **no seq** command removes the rule with the specified sequence number from the ACL. The **default seq** command also removes the specified rule.

The **seq** keyword is a command option used at the beginning of **deny (IPv6 Prefix List)** and **permit (IPv6 Prefix List)** commands that places a new rule between two existing rules.

Command Mode

IPv6-pfx Configuration

Command Syntax

```
no seq line_num
default seq line_num
```

Parameters

- *line_num* sequence number of rule to be deleted. Valid rule numbers range from 0 to 65535.

Example

- These commands remove rule 20 from the ***map1*** prefix list, then displays the resultant list.

```
switch(config)#ipv6 prefix-list map1
switch(config-ipv6-pfx)#no seq 20
switch(config-ipv6-pfx)#exit
switch(config)#show ipv6 prefix-list map1
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 30 permit 3:1bca:3ff2:634a::/64
seq 40 permit 3:1bca:1141:ab34::/64
switch(config)#
```

set (route-map configuration mode)

The **set** command specifies modifications to routes that are redistributed.

The **no set** and **default set** commands remove the **set** statement from the configuration mode route map clause by deleting the corresponding **set** statement from ***running-config***.

Command Mode

Route-Map-Configuration

Command Syntax

```
set CONDITION
no set CONDITION
default set CONDITION
```

Parameters

- *CONDITION* specifies the route modification parameter and value. Options include:
  - **as-path prepend** *path_name* BGP autonomous system path access list.
  - **community** *aa:nn* community number.
  - **community additive** Add to the existing community.
  - **community delete** Delete matching communities.
  - **community internet** Advertise to Internet community.
  - **community local-as** Do not send outside local AS.
  - **community no-advertise** Do not advertise to any peer.
  - **community no-export** Do not export to next AS.
  - **community none** Remove community attribute.
  - **community** *comm_number* community number. Value ranges from 0 to 4294967040.
  - **extcommunity additive** Add to the existing extcommunity.
  - **extcommunity delete** Delete matching extended communities.
  - **extcommunity none** Remove extended community attribute.
  - **extcommunity rt** *ASN:nn* Route Target extended community (AS:network number).
  - **extcommunity rt** *IP-address:nn* VPN extended community (IP address: network number).
  - **extcommunity soo** *ASN:nn* Site of origin ext. community (AS:network number).
  - **extcommunity soo** *IP-address:nn* Site of origin ext. community (IP address: network number).
  - **ip next-hop** *ipv4_address* next hop IPv4 address.
  - **ipv6 next-hop** *ipv6_address* next hop IPv6 address.
  - **local-preference** *preference_number* BGP local preference metric (0-4294967295).
  - **metric** *metric_number* route metric (0-4294967295).
  - **metric metric-type type-1** OSPF type 1 metric.
  - **metric metric-type type-2** OSPF type 2 metric.
  - **origin egp** BGP origin attribute.
  - **origin igp** BGP origin attribute.
  - **origin incomplete** BGP origin attribute.
  - **tag** *tag_number* route tag (0-4294967295).
  - **weight** *weight_value* BGP weight parameter (0 - 65535).

Example

- This command creates a route-map entry that sets the local preference metric to 100 on redistributed routes.

```
switch(config)#route-map map1
switch(config-route-map-map1)#set local-preference 100
switch(config-route-map-map1)#
```

show (ACL configuration modes)

The **show** command displays the ACL (Access Control List) contents:

- **show** or **show pending** – displays the list as modified in ACL configuration mode.
- **show active** – displays the list as stored in ***running-config***.
- **show comment** – displays the comment stored with the list.
- **show diff** – displays the modified and stored lists, with flags denoting the modified rules.

Exiting the ACL configuration mode stores all pending ACL changes to ***running-config***.

Command Mode

ACL Configuration
IPv6-ACL Configuration
Std-ACL Configuration
Std-IPv6-ACL Configuration
MAC-ACL Configuration

Command Syntax

```
show
show active
show comment
show diff
show pending
```

Examples

The examples in this section assume these ACL commands are entered as specified.

These commands are stored in running-config:

```
10 permit ip 10.10.10.0/24 any
20 permit ip any host 10.21.10.1
30 deny ip host 10.10.10.1 host 10.20.10.1
40 permit ip any any
50 remark end of list
```

The current edit session removed this command. This change is not yet stored to running-config:

```
20 permit ip any host 10.21.10.1
```

The current edit session added these commands ACL. They are not yet stored to running-config:

```
20 permit ip 10.10.0.0/16 any
25 permit tcp 10.10.20.0/24 any
45 deny pim 239.24.124.0/24 10.5.8.4/30
```

- This command displays the ACL, as stored in the configuration

```
switch(config-acl-test_1)#show active
IP Access List test_1
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.21.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        50 remark end of list
```

- This command displays the pending ACL, as modified in ACL Configuration Mode.

```
switch(config-acl-test_1)#show pending
IP Access List test_1
        10 permit ip 10.10.10.0/24 any
        20 permit ip 10.10.0.0/16 any
        25 permit tcp 10.10.20.0/24 any
        30 deny ip host 10.10.10.1 host 10.20.10.1
        40 permit ip any any
        45 deny pim 239.24.124.0/24 10.5.8.4/30
        50 remark end of list
```

- This command displays the difference between the saved and modified ACLs.

  Rules added to the pending list are denoted with a plus sign (+).
  Rules removed from the saved list are denoted with a minus sign (-)

```
switch(config-acl-test_1)#show diff
---
+++
@@ -1,7 +1,9 @@
 IP Access List test_1
         10 permit ip 10.10.10.0/24 any
-        20 permit ip any host 10.21.10.1
+        20 permit ip 10.10.0.0/16 any
+        25 permit tcp 10.10.20.0/24 any
         30 deny ip host 10.10.10.1 host 10.20.10.1
         40 permit ip any any
+        45 deny pim 239.24.124.0/24 10.5.8.4/30
```

show (route-map configuration mode)

The **show** command displays the route map as stored in ***running-config***. The display does not reflect changes to the route map made during the current editing session; those changes are displayed by exiting, then re-entering route-map configuration mode.

When the configuration contains multiple route maps with the same name and different sequence numbers or filter types, this command lists the contents of all route maps.

Command Mode

Route-Map-Configuration

Command Syntax

show

Example

- This command displays the ***map1*** route map, as stored in ***running-config***:

```
switch(config)#route-map map1
switch(config-route-map-map1)#show
route-map map1 permit 5
  Match clauses:
    match as 456
Set clauses:
route-map map1 permit 10
  Match clauses:
match ip next-hop 2.3.4.5
    match as-path path_2
  Set clauses:
    set local-preference 100
```

show ip access-lists

The **show ip access-list** command displays the contents of IPv4 and standard IPv4 access control lists (ACLs) on the switch. Use the **summary** option to display only the name of the lists and the number of lines in each list.

Command Mode

Privileged EXEC

Command Syntax

show ip access-list [***LIST***] [***SCOPE***]

Parameters

- ***LIST*** name of lists to be displayed. Selection options include:
  - <no parameter> all IPv4 ACLs are displayed.
  - *list_name* specified IPv4 ACL is displayed.
- ***SCOPE*** information displayed. Selection options include:
  - <no parameter> all rules in the specified lists are displayed.
  - **summary** the number of rules in the specified lists are displayed.

Examples

- This command displays all rules in ***test1*** IPv4 ACL.

```
switch#show ip access-list list2
IP Access List list2
        10 permit ip 10.10.10.0/24 any
        20 permit ip any host 10.20.10.1
        30 deny ip host 10.10.10.1 host 10.20.10.1
switch#
```

- This command displays the name of, and number of rules in, each list on the switch.

```
switch#show ip access-list summary
IPV4 ACL default-control-plane-acl
        Total rules configured: 12
        Configured on: control-plane
        Active on    : control-plane

IPV4 ACL list2
        Total rules configured: 3

IPV4 ACL test1
        Total rules configured: 6

Standard IPV4 ACL test_1
        Total rules configured: 1

IPV4 ACL test_3
        Total rules configured: 0

switch#
```

show ip prefix-list

The **show ip prefix-list** command displays all rules for the specified IPv4 prefix list. The command displays all IPv4 prefix list rules if a prefix list name is not specified.

Command Mode

EXEC

Command Syntax

show ip prefix-list [***DISPLAY_ITEMS***]

Parameters

- ***DISPLAY_ITEMS*** specifies the name of prefix lists for which rules are displayed. Options include:
  - <no parameter> all IPv4 prefix list rules are displayed.
  - *list_name* specifies the IPv4 prefix list for which rules are displayed.

Examples

- This command displays all rules in the route-one IPv4 prefix list:

```
switch>show ip prefix-list route-one
ip prefix-list route-one seq 10 deny 10.1.1.0/24
ip prefix-list route-one seq 20 deny 10.1.0.0/16
switch>
```

- This command displays all prefix list rules:

```
switch>show ip prefix-list
ip prefix-list route-one seq 10 deny 10.1.1.0/24
ip prefix-list route-one seq 20 deny 10.1.0.0/16
ip prefix-list route-two seq 10 deny 10.1.1.0/24 ge 26 le 30
ip prefix-list route-two seq 20 deny 10.1.0.0/16
ip prefix-list route-two seq 30 deny 3.3.3.3/32
ip prefix-list route-two seq 500 deny 1.1.1.0/24 ge 28 le 30
switch>
```

show ipv6 access-lists

The **show ipv6 access-list** command displays the contents of all IPv6 access control lists (ACLs) on the switch. Use the **summary** option to display only the name of the lists and the number of lines in each list.

Command Mode

Privileged EXEC

Command Syntax

show ipv6 access-list [***LIST***] [***SCOPE***]

Parameters

- ***LIST*** name of lists to be displayed. Selection options include:
  - <no parameter> all IPv6 ACLs are displayed.
  - *list_name* specified IPv6 ACL is displayed.
- ***SCOPE*** information displayed. Selection options include:
  - <no parameter> all rules in the specified lists are displayed.
  - **summary** the number of rules in the specified lists are displayed.

Examples

- This command displays all rules in test1 IPv6 ACL.

```
switch#show ipv6 access-list list2
IP Access List list2
        10 permit ipv6 3891:3c58:6300::/64 any
        20 permit ipv6 any host 2fe1:b468:024a::
        30 deny ipv6 host 3411:91c1:: host 4210:cc23:d2de:::
switch#
```

- This command displays the name of, and number of rules in, each list on the switch.

```
switch#show ipv6 access-list summary
IPV6 ACL list2
        Total rules configured: 3

IPV6 ACL test1
        Total rules configured: 6

IPV6 ACL test_1
        Total rules configured: 1

Standard IPV6 ACL test_3
        Total rules configured: 0

switch#
```

show ipv6 prefix-list

The **show ipv6 prefix-list** command displays all rules for the specified IPv6 prefix list. The command displays all IPv6 prefix lists if a prefix list name is not specified.

Command Mode

EXEC

Command Syntax

show ipv6 prefix-list [***DISPLAY_ITEMS***]

Parameters

- ***DISPLAY_ITEMS*** specifies the name of prefix lists for which rules are displayed. Options include:
  - <no parameter> all IPv6 prefix lists are displayed.
  - *list_name* specifies the IPv6 prefix list for which rules are displayed.

Examples

- This command displays all rules in the map1 IPv6 prefix list:

```
switch>show ipv6 prefix-list map1
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 20 permit 3:11b1:8fe4:1aac::/64
seq 30 permit 3:1bca:3ff2:634a::/64
seq 40 permit 3:1bca:1141:ab34::/64
switch>
```

- This command displays all prefix lists:

```
switch>show ipv6 prefix-list
ipv6 prefix-list map1
seq 10 permit 3:4e96:8ca1:33cf::/64
seq 15 deny 3:4400::/64
seq 20 permit 3:11b1:8fe4:1aac::/64
seq 30 permit 3:1bca:3ff2:634a::/64
seq 40 permit 3:1bca:1141:ab34::/64
ipv6 prefix-list FREDD
ipv6 prefix-list route-five
ipv6 prefix-list map2
seq 10 deny 10:1:1:1::/64 ge 72 le 80
seq 20 deny 10:1::/32
switch>
```

show mac access-lists

The **show mac access-list** command displays the contents of all MAC access control lists on the switch. Use the summary to display only the name of the lists and the number of lines in each list.

Command Mode

Privileged EXEC

Command Syntax

show mac access-lists [***LIST***] [***SCOPE***]

Parameters

- ***LIST*** name of lists to be displayed. Selection options include:
  - <no parameter> command displays all ACLs.
  - *list_name* command displays ACL specified by parameter
- ***SCOPE*** information displayed. Selection options include:
  - <no parameter> command displays all rules in specified lists.
  - **summary** command displays the number of rules in specified lists.

Examples

- This command displays all rules in ***mtest2*** MAC ACL.

```
switch#show mac access-list mlist2
IP Access List mlist2
        10 permit 1024.4510.F125 0.0.0 any aarp
        20 permit any 4100.4500.0000 0.FF.FFFF novell
        30 deny any any
switch#
```

- This command displays the name of, and number of, rules in, each list on the switch.

```
switch#show mac access-list summary
MAC ACL mlist1
        Total rules configured: 6

MAC ACL mlist2
        Total rules configured: 3

MAC ACL mlist3
        Total rules configured: 1

MAC ACL mlist4
        Total rules configured: 0

switch#
```

show route-map

The **show route-map** command displays the contents of the specified route maps. The command displays all route maps if an individual map is not specified.

Command Mode

EXEC

Command Syntax

show route-map [***MAP***]

Parameters

- *MAP* name of maps to be displayed. Selection options include:
  - <no parameter> command displays all ACLs.
  - *map_name* route map that the command displays.

Example

- This command displays the ***map1*** route map.

```
switch>show route-map map1
route-map map1 permit 5
  Match clauses:
    match as 456
Set clauses:
route-map map1 permit 10
  Match clauses:
match ip next-hop 2.3.4.5
    match as-path path_2
  Set clauses:
    set local-preference 100
```

statistics per-entry (ACL configuration modes)

The **statistics per-entry** command places the ACL in counting mode. An ACL is counting mode displays the number of instances each rule in the list matches an inbound packet and the elapsed time since the last match. The show access list commands display the statistics next to each rule in the ACL.

The **no statistics per-entry** and **default statistics per-entry** command places the ACL in non-counting mode.

Command Mode

ACL Configuration
IPv6-ACL Configuration
Std-ACL Configuration
Std-IPv6-ACL Configuration
MAC-ACL Configuration

Command Syntax

```
statistics per-entry
no statistics per-entry
default statistics per-entry
```

Examples

- This command places the test1 ACL in counting mode.

```
switch(config)#ip access-list test1
switch(config-acl-test1)#statistics per-entry
switch(config-acl-test1)#
```

- This command displays the ACL, with counter information, for an ACL in counting mode.

```
switch#show ip access-lists
IP Access List default-control-plane-acl [readonly]
     statistics per-entry
     10 permit icmp any any
     20 permit ip any any tracked [match 12041, 0:00:00 ago]
     30 permit ospf any any
     40 permit tcp any any eq ssh telnet www snmp bgp https [match 11, 1:41:07 ago]
     50 permit udp any any eq bootps bootpc snmp rip [match 78, 0:00:27 ago]
     60 permit tcp any any eq mlag ttl eq 255
     70 permit udp any any eq mlag ttl eq 255
     80 permit vrrp any any
     90 permit ahp any any
     100 permit pim any any
     110 permit igmp any any [match 14, 0:23:27 ago]
     120 permit tcp any any range 5900 5910
     130 permit tcp any any range 50000 50100
     140 permit udp any any range 51000 51100
```

Chapter 14

VRRP and VARP

A virtual IP (VIP) address is an IP address that does not directly connect to a specific interface. Inbound packets sent to a Virtual IP address are redirected to a physical network interface. VIPs supports connection redundancy by assigning the address to multiple switches. If one device becomes unavailable, packets sent to the address are still serviced by the functioning device.

Arista switches support virtual IP addresses through the Virtual Router Redundancy Protocol (VRRP) and the Virtual-ARP (VARP) feature. This chapter describes the Arista switch support of virtual IP addresses and contains these sections:

- Section 14.1: VRRP and VARP Conceptual Overview
- Section 14.2: VRRP and VARP Implementation Procedures
- Section 14.3: VRRP and VARP Implementation Examples
- Section 14.4: VRRP and VARP Configuration Commands

14.1 VRRP and VARP Conceptual Overview

14.1.1 VRRP

The Virtual Router Redundancy Protocol (VRRP) enables a group of routers to form a single *virtual router* to provide redundancy protection in an active-standby router configuration. The protocol defines a *virtual router* as an abstract object that is controlled through VRRP to act as a default router for hosts on a shared LAN.

A virtual router, also known as a *virtual router group*, is defined by a virtual router identifier (VRID) and a virtual IP address. A virtual router's mapping of VRID and IP address must be consistent among all switches implementing the virtual router group. Two virtual routers cannot be assigned the same VRID, even when they are on different VLANs. A virtual router's scope is restricted to a single LAN.

A LAN may contain multiple virtual routers for distributing traffic. Each virtual router on a LAN is assigned a unique VRID. A switch may be configured with virtual routers among multiple LANs.

VRRP uses priority ratings to assign Master or Backup roles for each VRRP router configured for a virtual router group. The Master router sends periodic VRRP Advertisement messages along the LAN and forwards packets received by the virtual router to their destination. Backup routers are inactive but are available to assume Master router duties when the current Master fails.

A VRRP can be configured to allow VRRP routers with higher priority to take over Master router duties. Alternatively, the group can be configured to prevent a router from preemptively assuming the Master role. A VRRP router is always assigned the Master of any virtual router configured with the address owned by the VRRP router, regardless of the preemption prevention setting.

14.1.2 VARP

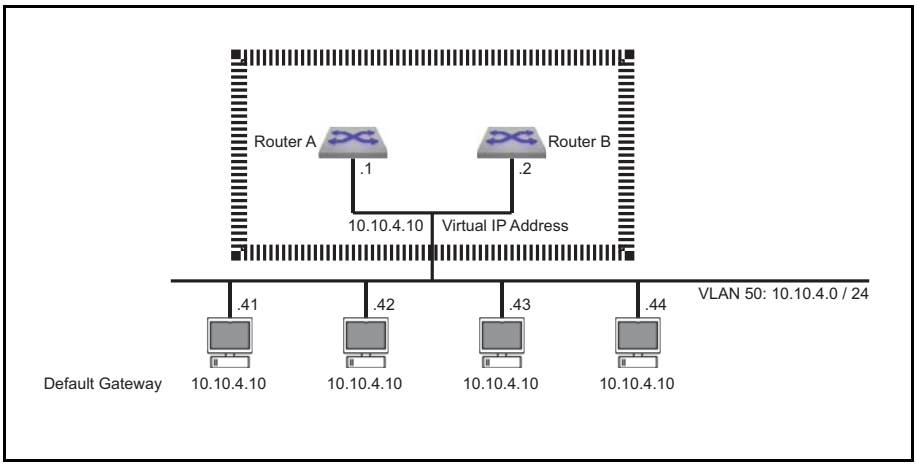
Virtual-ARP (VARP) allows multiple switches to simultaneously route packets from a common IP address in an active-active router configuration. Each switch is configured with the same set of virtual IP addresses on corresponding VLAN interfaces and a common virtual MAC address. In MLAG configurations, VARP is preferred over VRRP because VARP does not require traffic to traverse the peer-link to the master router as VRRP would.

A maximum of 500 virtual IP addresses can be assigned to a VLAN interface. All virtual addresses on all VLAN interfaces resolve to the same virtual MAC address.

VARP functions by having each switch respond to ARP and GARP requests for the configured router IP address with the virtual MAC address. The virtual MAC address is only for inbound packets and never used in the source field of outbound packets.

When ***ip routing*** is enabled, packets to the virtual MAC address are routed to the next hop destination.

Figure 14-1 VARP Configuration



14.2 VRRP and VARP Implementation Procedures

This section contains the following configuration instructions:

- Section 14.2.1: VRRP Configuration
- Section 14.2.2: VARP Configuration

14.2.1 VRRP Configuration

Implementing a virtual router consists of configuration and enabling commands. A virtual router is typically configured before it is enabled; this ensures that the VRRP router operates as required before its priority settings immediately make it the master virtual router. Because a virtual router is enabled by assigning it a primary address, it is normally performed after all other configuration step tasks.

The **no vrrp** command removes all vrrp commands for the specified virtual router from ***running-config***.

14.2.1.1 Virtual Router Configuration

Most configuration tasks are optional because all mandatory parameters have a default value. The following virtual router parameters are configurable:

- Router priority (default = 100)
- Preemption option (default is enabled)
- Advertisement timer (default = one second)
- Description (optional parameter)
- Authentication (optional parameter)
- Secondary IP addresses (optional parameter)

Designating the Master and Backup Router

The VRRP routers within a virtual router group determine the Master router through priority settings. Priority values range from 254 (highest priority) to 1 (lowest priority). Priority is either set by a CLI command or is assigned the default value of 100. A switch specifies priority settings for each of its virtual routers.

Preemption mode determines when a VRRP router with a higher priority rating becomes the Master router. If preemption is enabled, the VRRP router with the highest priority immediately becomes the Master router. If preemption is disabled, a VRRP router with a higher priority value does not become the Master router unless the current Master becomes unavailable; this is applicable when a new VRRP router becomes available on the LAN or VRRP router's priority value changes for the virtual router.

The **vrrp priority** command configures the switch's priority setting for the specified virtual router.

Example

- This command sets the priority value of 250 for the virtual router with VRID 15 on VLAN 20.

```
switch(config-if-vl20)#vrrp 15 priority 250
switch(config-if-vl20)#
```

The **vrrp preempt** command controls the preempt mode setting of the specified virtual router. By default, preempt mode is enabled.

Examples

- This command disables preempt mode for the virtual router 15 on VLAN 20.

```
switch(config-if-vl20)#no vrrp 15 preempt
switch(config-if-vl20)#
```

- This command enables preempt mode for the virtual router 30 on VLAN 20.

```
switch(config-if-vl20)#vrrp 30 preempt
switch(config-if-vl20)#
```

The **vrrp preempt delay** command configures a period between an event that elevates a switch to master vrrp router status and the switch's assumption of master vrrp router role. Command options configure delays during normal operation and after a switch reboot.

Advertisement Timer

The Master router sends periodic VRRP Advertisement messages to other VRRP routers. The **vrrp timers advertise** command specifies the interval between successive advertisement message transmissions.

The advertisement interval also defines the timeout that determines when the switch assumes the Master router role. This timeout interval is three times the advertisement interval.

Example

- This command sets the advertisement interval of 10 seconds for virtual router 35 on VLAN 100.

```
switch(config-if-vl100)#vrrp 35 timers advertise 10
switch(config-if-vl100)#
```

Description

The **vrrp description** command associates a text string to the specified virtual router. The maximum string length is 80 characters. The string has no functional impact on the virtual router.

Example

- This command associates the text string ***Laboratory Router*** to virtual router 15 on VLAN 20.

```
switch(config-if-vl20)#vrrp 15 description Laboratory Router
switch(config-if-vl20)#
```

Authentication

VRRP authentication validates VRRP advertisement packets that the switch receives from other VRRP routers in a specified virtual router group. When a virtual router uses authentication, all VRRP routers in the group must use the same authentication parameters.

The **vrrp authentication** command configures virtual router authentication parameters for the specified virtual router.

Example

- This command implements plain-text authentication, using 12345 as the key, for virtual router 40 on VLAN 100.

```
switch(config-if-vl100)#vrrp 40 authentication text 12345
switch(config-if-vl100)#
```

Secondary Addresses

The **vrrp ip secondary** command assigns a secondary IP address to a virtual router. Secondary addresses are optional; a virtual router's configuration may include more than one secondary address command. The primary and secondary address list must be identical for all switches in a virtual router group.

A primary IP address is assigned to a virtual router with the **vrrp ip** command (Section 14.2.1.2).

Example

- This command assigns the IP address of 10.2.4.5 as the secondary IP address for the virtual router 15 on VLAN 20

```
switch(config-if-vl20)#vrrp 15 ip 10.2.4.5 secondary
switch(config-if-vl20)#
```

14.2.1.2 Virtual Router Enabling and the Primary IP address

The **vrrp ip** command configures the primary IP address of the specified virtual router and enables the virtual router if the primary address is contained within the configuration mode interface's IP address subnet. A virtual router's configuration may contain only one primary IP address assignment command; subsequent **vrrp ip** commands reassign the virtual router's primary IP address.

Example

- This command enables virtual router group 15 (VRID) on VLAN 20 and assigns 10.1.1.5 as the virtual router's primary address.

```
switch(config-if-vl20)#vrrp 15 ip 10.1.1.5
switch(config-if-vl20)#
```

14.2.1.3 VRRP Disabling and Shutdown

The **vrrp shutdown** command places the switch in stopped state for the specified virtual router. While in stopped state, the switch cannot act as a Master or backup router for the virtual router group. The ***no vrrp shutdown*** command changes the switch's virtual router state to ***backup*** or ***master*** if the virtual router is properly configured.

Example

- This command places the switch in stopped mode for virtual router 24 on VLAN 20.

```
switch(config-if-vl20)#vrrp 24 shutdown
switch(config-if-vl20)#
```

- This command moves the switch out of stopped mode for virtual router 24 on VLAN 20.

```
switch(config-if-vl20)#no vrrp 24 shutdown
switch(config-if-vl20)#
```

The **no vrrp** and **no vrrp ip** commands delete the specified virtual IP address from the interface. Additionally, the **no vrrp** command removes all residual VRRP commands for the virtual router.

- This command removes all vrrp configuration commands for virtual router 10 on VLAN 15.

```
switch(config-if-vl15)#no vrrp 10
switch(config-if-vl15)#
```

- This command disables virtual router 25 on VLAN 20 and removes the primary IP address from its configuration.

```
switch(config-if-vl20)#no vrrp 25 ip 10.1.1.5
switch(config-if-vl20)#
```

14.2.2 VARP Configuration

Implementing VARP consists of assigning virtual IP addresses to VLAN interfaces and configuring a virtual MAC address.

Virtual IP Addresses

The **ip virtual-router address** command assigns a virtual IP address to the configuration mode interface. The virtual router's IP address on a LAN can be used as the default first hop router by end-hosts. The IP address should be in the subnet of the IP address assigned to the interface.

Example

- This command configures the Switch Virtual Interface (SVI) and a virtual IP address for VLAN 4094.

```
Switch(config)#interface vlan 4094
Switch(config-if-Vl4094)#ip address 10.0.0.2/24
Switch(config-if-Vl4094)#ip virtual-router address 10.0.0.6
Switch(config-if-Vl4094)#exit
Switch(config)#
```

Virtual MAC Address

The **ip virtual-router mac-address** command assigns a virtual MAC address to the switch. The switch maps all virtual router IP addresses to this MAC address. The address is receive-only; the switch never sends packets with this address as the source.

When the destination MAC of a packet destined to a remote network matches the virtual MAC address, the MLAG peer forwards the traffic to the next hop destination. Each MLAG peer must have the same routes available, either though static configuration or learned through a dynamic routing protocol.

Example

- This command configures a virtual MAC address.

```
Switch(config)#ip virtual-router mac-address 001c.7300.0099
Switch(config)#
```

Virtual MAC Address

To display the virtual router MAC and IP addresses, enter the **show ip virtual-router** command.

Example

- This command displays the virtual router addresses assigned on the switch.

```
switch>show ip virtual-router
IP virtual router is configured with MAC address: 24cd.5a29.cc31
Interface  IP Address        Virtual IP Address   Status            Protocol
Vlan15     10.1.1.3/24       10.1.1.15            up                up
Vlan15     10.1.1.3/24       10.1.1.16            up                up
Vlan15     10.1.1.3/24       10.1.1.17            up                up
Vlan20     10.12.1.6/24      10.1.1.51            up                up
Vlan20     10.12.1.6/24      10.1.1.53            up                up
Vlan20     10.12.1.6/24      10.1.1.55            up                up
switch>
```

14.3 VRRP and VARP Implementation Examples

This section contains the following example set:

- Section 14.3.1: VRRP Examples
- Section 14.3.2: VARP Example

14.3.1 VRRP Examples

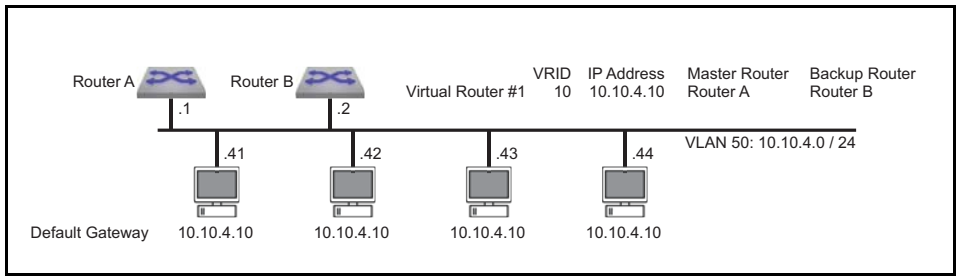
This section provides code that implements three VRRP configurations:

- Example 1 configures two switches in a single virtual router group. This implementation protects the LAN against the failure of one router.
- Example 2 configures two switches into two virtual routers within a single LAN. This implementation protects the LAN against the failure of one router and balances traffic between the routers.
- Example 3 configures three switches to implement virtual routers on two LANs. Each LAN contains two virtual routers. One switch is configured into four virtual routers – two on each LAN.

14.3.1.1 VRRP Example 1: One Virtual Router on One LAN

Figure 14-2 displays the Example 1 network. Two switches are configured as VRRP routers to form one virtual router.

Figure 14-2 VRRP Example 1 Network Diagram



The following code configures the first switch (Router A) as the master router and the second switch (Router B) as a backup router for virtual router 10 on VLAN 50. Router A becomes the Master virtual router by setting its priority at 200; Router B maintains the default priority of 100. The advertisement interval is three seconds on both switches. Priority preemption is enabled by default.

Switch code that implements Router A on the first switch

```
Switch-A(config)#interface vlan 50
Switch-A(config-if-vl50)#ip address 10.10.4.1/24
Switch-A(config-if-vl50)#no vrrp 10
Switch-A(config-if-vl50)#vrrp 10 priority 200
Switch-A(config-if-vl50)#vrrp 10 timers advertise 3
Switch-A(config-if-vl50)#vrrp 10 ip 10.10.4.10
Switch-A(config-if-vl50)#exit
```

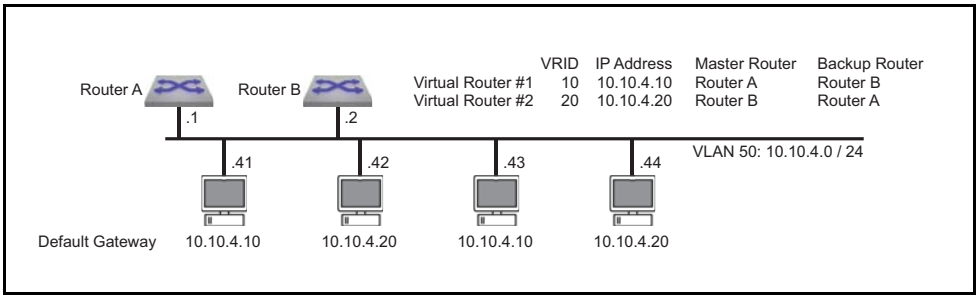
Switch code that implements Router B on the second switch

```
Switch-B(config)#interface vlan 50
Switch-B(config-if-vl50)#ip address 10.10.4.2/24
Switch-B(config-if-vl50)#no vrrp 10
Switch-B(config-if-vl50)#vrrp 10 timers advertise 3
Switch-B(config-if-vl50)#vrrp 10 ip 10.10.4.10
Switch-B(config-if-vl50)#exit
```

14.3.1.2 VRRP Example 2: Two Virtual Routers on One LAN

Figure 14-3 displays Example 2. Two switches are configured as VRRP routers to form two virtual routers on one LAN. Using two virtual routers distributes the LAN traffic between the switches.

Figure 14-3 VRRP Example 2 Network Diagram



The following code configures two switches as a master and a backup router for two virtual routers on VLAN 50.

- Router A is the master for virtual router 10 and backup for virtual router 20.
- Router B is the master for virtual router 20 and backup for virtual router 10.
- VRRP advertisement interval is 3 seconds on virtual router 10 and 5 seconds on virtual router 20.
- Priority preemption is enabled by default for both virtual routers.

Switch code that implements Router A on the first switch

```
Switch-A(config)#interface vlan 50
Switch-A(config-if-vl50)#ip address 10.10.4.1/24
Switch-A(config-if-vl50)#no vrrp 10
Switch-A(config-if-vl50)#vrrp 10 priority 200
Switch-A(config-if-vl50)#vrrp 10 timers advertise 3
Switch-A(config-if-vl50)#vrrp 10 ip 10.10.4.10
Switch-A(config-if-vl50)#no vrrp 20
Switch-A(config-if-vl50)#vrrp 20 timers advertise 5
Switch-A(config-if-vl50)#vrrp 20 ip 10.10.4.20
Switch-A(config-if-vl50)#exit
```

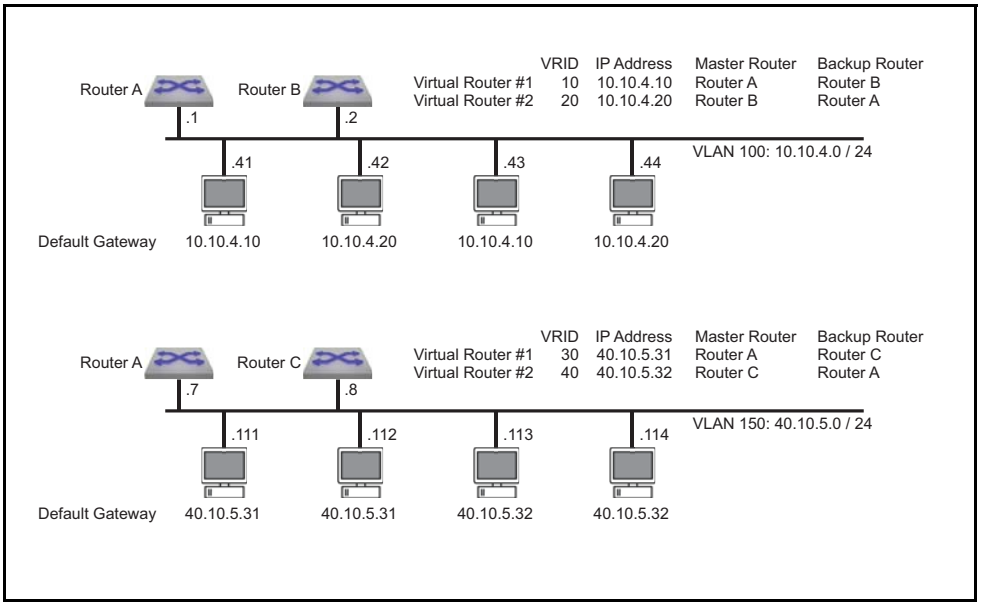
Switch code that implements Router B on the second switch

```
Switch-B(config)#interface vlan 50
Switch-B(config-if-vl50)#ip address 10.10.4.2/24
Switch-B(config-if-vl50)#no vrrp 10
Switch-B(config-if-vl50)#vrrp 10 timers advertise 3
Switch-B(config-if-vl50)#vrrp 10 ip 10.10.4.10
Switch-B(config-if-vl50)#no vrrp 20
Switch-B(config-if-vl50)#vrrp 20 priority 200
Switch-B(config-if-vl50)#vrrp 20 timers advertise 5
Switch-B(config-if-vl50)#vrrp 20 ip 10.10.4.20
Switch-B(config-if-vl50)#exit
```

14.3.1.3 VRRP Example 3: Two Virtual Routers on Two LANs

Figure 14-4 displays Example 3. Three switches are configured as VRRP routers to form four virtual router groups – two groups on each of two LANs.

Figure 14-4 VRRP Example 3 Network Diagram



The following code configures the three switches as follows:

- Router A is the master for virtual router 10 and backup for virtual router 20 on VLAN 100.
- Router A is the master for virtual router 30 and backup for virtual router 40 on VLAN 150.
- Router B is the master for virtual router 20 and backup for virtual router 10 on VLAN 100.
- Router C is the master for virtual router 40 and backup for virtual router 30 on VLAN 150.
- VRRP advertisement interval is set to one second on all virtual routers.
- Priority preemption is disabled on all virtual routers.

Switch code that implements Router A on the first switch

```
Switch-A(config)#interface vlan 100
Switch-A(config-if-vl100)#ip address 10.10.4.1/24
Switch-A(config-if-vl100)#no vrrp 10
Switch-A(config-if-vl100)#vrrp 10 priority 200
Switch-A(config-if-vl100)#no vrrp 10 preempt
Switch-A(config-if-vl100)#vrrp 10 ip 10.10.4.10
Switch-A(config-if-vl100)#no vrrp 20
Switch-A(config-if-vl100)#no vrrp 20 preempt
Switch-A(config-if-vl100)#vrrp 20 ip 10.10.4.20
Switch-A(config-if-vl100)#interface vlan 150
Switch-A(config-if-vl150)#ip address 40.10.5.7/24
Switch-A(config-if-vl150)#no vrrp 30
Switch-A(config-if-vl150)#vrrp 30 priority 200
Switch-A(config-if-vl150)#no vrrp 30 preempt
Switch-A(config-if-vl150)#vrrp 30 ip 40.10.5.31
Switch-A(config-if-vl150)#no vrrp 40
Switch-A(config-if-vl150)#no vrrp 40 preempt
Switch-A(config-if-vl150)#vrrp 40 ip 40.10.5.32
Switch-A(config-if-vl150)#exit
```

Switch code that implements Router B on the second switch

```
Switch-B(config)#interface vlan 100
Switch-B(config-if-vl100)#ip address 10.10.4.2/24
Switch-B(config-if-vl100)#no vrrp 10
Switch-B(config-if-vl100)#no vrrp 10 preempt
Switch-B(config-if-vl100)#vrrp 10 ip 10.10.4.10
Switch-B(config-if-vl100)#no vrrp 20
Switch-B(config-if-vl100)#vrrp 20 priority 200
Switch-B(config-if-vl100)#no vrrp 20 preempt
Switch-B(config-if-vl100)#vrrp 20 ip 10.10.4.20
Switch-B(config-if-vl100)#exit
```

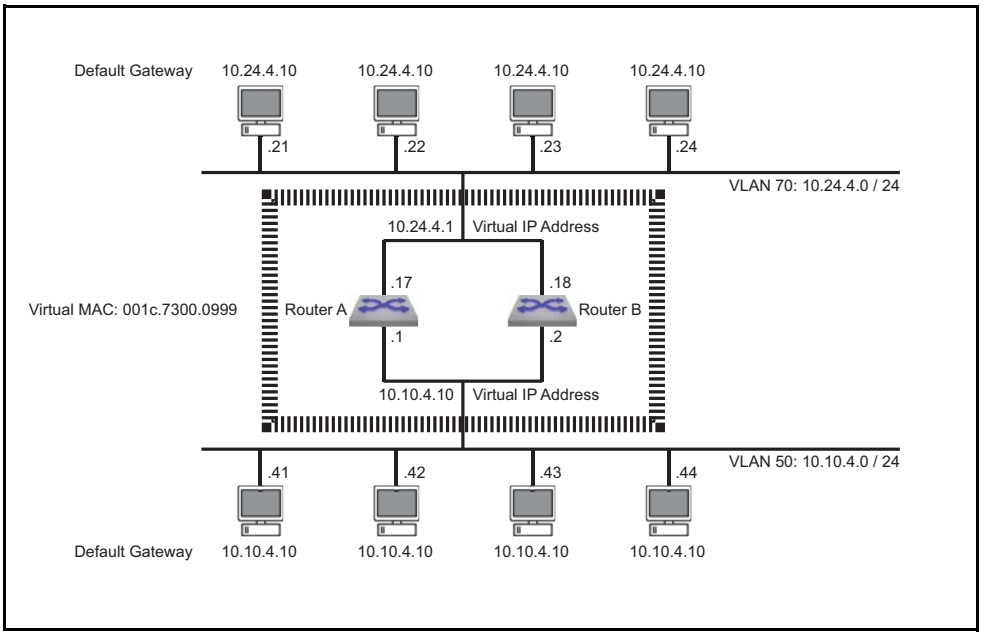
Switch code that implements Router C on the third switch

```
Switch-C(config)#interface vlan 150
Switch-C(config-if-vl150)#ip address 40.10.5.8/24
Switch-C(config-if-vl150)#no vrrp 30
Switch-C(config-if-vl150)#no vrrp 30 preempt
Switch-C(config-if-vl150)#vrrp 30 ip 40.10.5.31
Switch-C(config-if-vl150)#no vrrp 40
Switch-C(config-if-vl150)#vrrp 40 priority 200
Switch-C(config-if-vl150)#no vrrp 40 preempt
Switch-C(config-if-vl150)#vrrp 40 ip 40.10.5.32
Switch-C(config-if-vl150)#exit
```

14.3.2 VARP Example

This section provides code that implements a VARP configuration. Figure 14-5 displays the Example 1 network. Two switches in an MLAG domain are configured as VARP routers.

Figure 14-5 VARP Example Network Diagram



The following code configures 10.10.4.10 as the virtual IP address for VLAN 50, 10.24.4.1 as the virtual IP address for VLAN 70, and 001c.7300.0999 as the virtual MAC address on both switches.

Switch code that implements VARP on the first switch

```
Switch-A(config)#ip virtual-router mac-address 001c.7300.0999
Switch-A(config)#interface vlan 50
Switch-A(config-if-vl50)#ip address 10.10.4.1/24
Switch-A(config-if-vl50)#ip virtual-router address 10.10.4.10
Switch-A(config-if-vl50)#interface vlan 70
Switch-A(config-if-vl70)#ip address 10.24.4.17/24
Switch-A(config-if-vl70)#ip virtual-router address 10.24.4.1
Switch-A(config-if-vl70)#exit
```

Switch code that implements VARP on the second switch

```
Switch-B(config)#ip virtual-router mac-address 001c.7300.0999
Switch-B(config)#interface vlan 50
Switch-B(config-if-vl50)#ip address 10.10.4.2/24
Switch-B(config-if-vl50)#ip virtual-router address 10.10.4.10
Switch-B(config-if-vl50)#interface vlan 70
Switch-B(config-if-vl70)#ip address 10.24.4.18/24
Switch-B(config-if-vl70)#ip virtual-router address 10.24.4.1
Switch-B(config-if-vl70)#exit
```

14.4 VRRP and VARP Configuration Commands

This section contains descriptions CLI commands that support VRRP and VARP.

Interface Configuration Commands – VLAN Interface

- ip virtual-router mac-address . . . . . Page 544
- ip virtual-router mac-address advertisement-interval . . . . . Page 545

Interface Configuration Commands – Ethernet, Port Channel, and VLAN Interfaces

- ip virtual-router address . . . . . Page 543
- no vrrp . . . . . Page 546
- vrrp authentication . . . . . Page 550
- vrrp description . . . . . Page 551
- vrrp ip . . . . . Page 552
- vrrp ip secondary . . . . . Page 553
- vrrp preempt . . . . . Page 554
- vrrp preempt delay . . . . . Page 555
- vrrp priority . . . . . Page 557
- vrrp shutdown . . . . . Page 558
- vrrp timers advertise . . . . . Page 559
- vrrp track . . . . . Page 560

Privileged EXEC Commands

- show ip virtual-router . . . . . Page 547
- show vrrp . . . . . Page 548

ip virtual-router address

The **ip virtual-router address** command assigns a virtual IP address to the configuration mode interface. The virtual router's IP address on a LAN can be used as the default first hop router by end-hosts. The IP address should be in the subnet of the IP address assigned to the interface.

A maximum of 500 virtual IP address can be assigned to a VLAN interface. All virtual addresses on all VLAN interfaces resolve to the same virtual MAC address configured through the **ip virtual-router mac-address** command.

This command is typically used in MLAG configurations to create identical virtual routers on switches connected to the MLAG domain through an MLAG.

The **no ip virtual-router address** and **default ip virtual-router address** commands removes the specified virtual IP address from the configuration mode interface by deleting the corresponding **ip virtual-router address** command from ***running-config***. If the command does not specify an address, all virtual IP addresses are removed from the interface.

Command Mode

Interface-VLAN Configuration

Command Syntax

```
ip virtual-router address net_addr
no ip virtual-router address [net_addr]
default ip virtual-router address [net_addr]
```

Parameters

- *net_addr* network IP address. Valid entry formats include CIDR and address-mask. ***Running-config*** stores value in CIDR notation.

Examples

- This command configures the Switch Virtual Interface (SVI) and a virtual IP address for VLAN 4094.

```
Switch(config)#interface vlan 4094
Switch(config-if-Vl4094)#ip address 10.0.0.2/24
Switch(config-if-Vl4094)#ip virtual-router address 10.0.0.6
Switch(config-if-Vl4094)#exit
Switch(config)#
```

ip virtual-router mac-address

The **ip virtual-router mac-address** command assigns a virtual MAC address to the switch. The switch maps all virtual router IP addresses to this MAC address. The address is receive-only; the switch never sends packets with this address as the source. The virtual router is not configured on the switch until this virtual mac-address is assigned.

This command is typically used in MLAG configurations to create identical virtual routers on switches connected to the MLAG domain through an MLAG. When the destination MAC of a packet destined to a remote network matches the virtual MAC address, the MLAG peer forwards the traffic to the next hop destination. Each MLAG peer must have the same routes available, either though static configuration or learned through a dynamic routing protocol.

The **no ip virtual-router mac-address** command removes a virtual MAC address from the interface by deleting the corresponding **ip virtual-router mac-address** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

ip virtual-router mac-address *mac_addr*
no ip virtual-router mac address [*mac_addr*]

Parameters

- *mac_addr* MAC IP address (dotted hex notation). Select an address that will not otherwise appear on the switch.

Examples

- This command configures a virtual MAC address.

```
Switch(config)#ip virtual-router mac-address 001c.7300.0099
Switch(config)#
```

ip virtual-router mac-address advertisement-interval

The **ip virtual-router mac-address advertisement interval** command specifies the period between the transmission of consecutive gratuitous ARP requests that contain the virtual router mac address for each virtual-router IP address configured on the switch. The default period is 30 seconds.

The **no ip virtual-router mac-address advertisement-interval** command restores the default period of 30 seconds by removing the **ip virtual-router mac-address advertisement-interval** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
ip virtual-router mac-address advertisement-interval period
no ip virtual-router mac-address advertisement-interval
default ip virtual-router mac-address advertisement-interval
```

Parameters

- *period* advertisement interval (seconds). Values range from 0 to 86400. Default is 30.

Examples

- This command configures a MAC address advertisement interval of one minute (60 seconds).

```
Switch(config)#ip virtual-router mac-address advertisement-interval 60
Switch(config)#
```

no vrrp

The **no vrrp** command removes all vrrp configuration commands for the specified virtual router on the configuration mode interface. The default vrrp command also reverts vrrp configuration parameters to default settings by removing the corresponding **vrrp** commands.

Commands removed by the **no vrrp** command include:

- **vrrp authentication**
- **vrrp description**
- **vrrp ip**
- **vrrp ip secondary**
- **vrrp preempt**
- **vrrp preempt delay**
- **vrrp priority**
- **vrrp shutdown**
- **vrrp timers advertise**

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
no vrrp group
default vrrp group
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.

Examples

- This command removes all vrrp configuration commands for virtual router group 10 on VLAN 15.

```
switch(config)#interface vlan 15
switch(config-if-vl15)#no vrrp 10
switch(config-if-vl15)#
```

show ip virtual-router

The **show ip virtual-router** command displays the virtual MAC address assigned to the switch and all virtual IP addresses assigned to each VLAN interface.

Command Mode
EXEC

Command Syntax
show ip virtual-router

Messages

- **IP virtual router is not configured** a virtual MAC address is not assigned to the switch.
- **No interface with virtual IP address** no virtual IP addresses are assigned to any VLAN interfaces.

Examples

- This command displays a table of information for VRRP groups on the switch.

```
switch>show ip virtual-router
IP virtual router is configured with MAC address: 24cd.5a29.cc31
Interface  IP Address        Virtual IP Address   Status            Protocol
Vlan15     10.1.1.3/24       10.1.1.15            up                up
Vlan15     10.1.1.3/24       10.1.1.16            up                up
Vlan15     10.1.1.3/24       10.1.1.17            up                up
Vlan20     10.12.1.6/24      10.1.1.51            up                up
Vlan20     10.12.1.6/24      10.1.1.53            up                up
Vlan20     10.12.1.6/24      10.1.1.55            up                up
switch>
```

- This command generates a response that indicates a virtual MAC address is not assigned to the switch.

```
switch>show ip virtual-router
IP virtual router is not configured
switch>
```

show vrrp

The **show vrrp interface** command displays the status of configured Virtual Router Redundancy Protocol (VRRP) groups on a specified interface. Parameter options control the amount and formatting of the displayed information.

Command Mode

EXEC

Command Syntax

show vrrp interface ***[INTERFACE_GROUP]*** [***INFO_LEVEL***] **[*STATES*]**

Parameters

- ***INTERFACE_GROUP*** specifies groups for which command displays status. When the parameter is omitted or specifies only an interface, the group list is filtered by the ***STATES*** parameter.
  - — <no parameter> all groups.
  - — **ethernet** *e_num* all groups on specified Ethernet interface.
  - — **loopback** *l_num* all groups on specified loopback interface.
  - — **management** *m_num* all groups on specified management interface.
  - — **port-channel** *p_num* all groups on specified port channel interface.
  - — **vlan** *v_num* all groups on specified VLAN interface.
  - — **ethernet** *e_num* **group** *group_num* specified group on specified Ethernet interface.
  - — **loopback** *l_num* **group** *group_num* specified group on specified loopback interface.
  - — **management** *m_num* **group** *group_num* specified group on specified management interface.
  - — **port-channel** *p_num* **group** *group_num* specified group on specified port channel interface.
  - — **vlan** *v_num* **group** *group_num* specified group on specified VLAN interface.
- ***INFO_LEVEL*** Specifies format and amount of displayed information. Options include:
  - — <no parameter> displays a block of data for each VRRP group.
  - — **brief** displays a single table that lists information for all VRRP groups.
- ***STATES*** Specifies the groups, by VRRP router state, that are displayed. Parameter is not available when ***INTERFACE_GROUP*** specifies one group. Options include:
  - — <no parameter> displays data for groups in the ***master*** or ***backup*** states.
  - — **all** displays all groups, including groups in the ***stopped*** and ***interface down*** states.

Examples

- This command displays a table of information for VRRP groups on the switch.

```
switch>show vrrp brief
Port       Group Prio Time   Own  State    MaIp              GrIp
Vlan1006   3     100  3609       Backup   127.38.10.2       127.38.10.1
Vlan1010   1     100  3609       Backup   128.44.5.3        128.44.5.1
Vlan1014   2     100  3609       Backup   127.16.14.2       127.16.14.1
```

- This command displays data blocks for all VRRP groups on VLAN 46, regardless of the VRRP state.

```
switch>show vrrp interface vlan 1006 all
Vlan46 - Group 3
  State is Backup
  Virtual IP address is 127.38.10.1
  Virtual MAC address is 0000.5e00.0103
  Advertisement interval is 1.000s
  Preemption is enabled
  Priority is 100
  Master Router is 127.38.10.2, priority is 100
  Master Advertisement interval is 1.000s
  Master Down interval is 3.609s

Vlan46 - Group 8
  State is Backup
  Virtual IP address is 128.44.5.1
  Virtual MAC address is 0000.5e00.0101
  Advertisement interval is 1.000s
  Preemption is enabled
  Priority is 100
  Master Router is 172.22.10.3, priority is 100
  Master Advertisement interval is 1.000s
  Master Down interval is 3.609s
```

- This command displays data for all VRRP group 2 on VLAN 1014.

```
switch>show vrrp interface vlan 1014 group 2
Vlan1014 - Group 2
  State is Master
  Virtual IP address is 172.22.14.1
  Virtual MAC address is 0000.5e00.0102
  Advertisement interval is 1.000s
  Preemption is enabled
  Preemption delay is 0.000s
  Preemption reload delay is 0.000s
  Priority is 100
  Master Router is 172.22.14.3 (local), priority is 100
  Master Advertisement interval is 1.000s
  Master Down interval is 3.609s
switch>
```

vrrp authentication

The **vrrp authentication** command configures parameters the switch uses to authenticate virtual router packets it receives from other VRRP routers in the group.

The **no vrrp authentication** and **default vrrp authentication** commands disable VRRP authentication of packets from the specified virtual router by removing the corresponding **vrrp authentication** command from *running-config*. The **no vrrp** command also removes the **vrrp authentication** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group authentication AUTH_PARAMETER
no vrrp group authentication
default vrrp group authentication
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- ***AUTH_PARAMETER*** encryption level and authentication key used by router. Options include:
  - **text** *text_key* plain-text authentication, *text_key* is text.
  - *text_key* plain-text authentication, *text_key* is text.
  - **ietf-md5 key-string 0** *text_key* IP authentication of MD5 key hash, *text_key* is text.
  - **ietf-md5 key-string** *text_key* IP authentication of MD5 key hash, *text_key* is text.
  - **ietf-md5 key-string 7** *coded_key* IP authentication of MD5 key hash, *coded_key* is MD5 hash.

Examples

- This command implements plain-text authentication, using 12345 as the key, for virtual router 40 on VLAN 100.

```
switch(config)#interface vlan 100
switch(config-if-vl100)#vrrp 40 authentication text 12345
switch(config-if-vl100)#
```

- This command implements ietf-md5 authentication, using 12345 as the key.

```
switch(config-if-vl100)#vrrp 40 authentication ietf-md5 key-string 0 12345
switch(config-if-vl100)#
```

- This command implements ietf-md5 authentication, using 12345 as the key. The key is entered as the MD5 hash equivalent of the text string.

```
switch(config-if-vl100)#vrrp 40 authentication ietf-md5 key-string 7
EA3TUPxdddFCLYT8mb+kxw==
switch(config-if-vl100)#
```

vrrp description

The **vrrp description** command associates a text string to a VRRP virtual router on the configuration mode interface. The string has no functional impact on the virtual router. The maximum length of the string is 80 characters.

The **no vrrp description** and **default vrrp description** commands remove the text string association from the VRRP virtual router by deleting the corresponding **vrrp description** command from *running-config*. The **no vrrp** command also removes the **vrrp description** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group description label_text
no vrrp group description
default vrrp group description
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *label_text* text that describes the virtual router. Maximum string length is 80 characters.

Example

- This command associates the text string ***Laboratory Router*** to virtual router 15 on VLAN 20.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 15 description Laboratory Router
switch(config-if-vl20)#
```

vrrp ip

The **vrrp ip** command configures the primary IP address for the specified VRRP virtual router. The command also activates the virtual router if the primary address is contained in the interface's subnet. A VRRP virtual router's configuration may contain only one primary IP address assignment command; subsequent **vrrp ip** commands replace the existing primary address assignment.

The **vrrp ip secondary** command assigns a secondary IP address to the VRRP virtual router.

The **no vrrp ip** and **default vrrp ip** commands disable the VRRP virtual router and deletes the primary IP address by removing the corresponding **vrrp ip** statement from ***running-config.*** The **no vrrp** command also removes the **vrrp ip** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group ip ipv4_address
no vrrp group ip ipv4_address
default vrrp group ip ipv4_address
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *ipv4_address* IPv4 address of the virtual router.

Related Commands

vrrp ip secondary

Example

- This command enables virtual router 15 on VLAN 20 and designates 10.1.1.5 as the virtual router's primary address.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 15 ip 10.1.1.5
switch(config-if-vl20)#
```

vrrp ip secondary

The **vrrp ip secondary** command assigns a secondary IP address to the specified virtual router. Secondary IP addresses are an optional virtual router parameter. A virtual router may contain multiple secondary address commands. The IP address list must be identical for all VRRP routers in a virtual router group.

The virtual router is assigned a primary IP address with the **vrrp ip** command.

The **no vrrp ip secondary** and **default vrrp ip secondary** commands remove the secondary IP address for the specified VRRP virtual router by deleting the corresponding **vrrp ip secondary** statement from *running-config*. The **no vrrp** command also removes all **vrrp secondary** commands for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group ip ipv4_addr secondary
no vrrp group ip ipv4_addr secondary
default vrrp group ip ipv4_addr secondary
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *ipv4_addr* secondary IPv4 address of the virtual router.

Related Commands

vrrp ip

Example

- This command assigns the IP address of 10.2.4.5 as the secondary IP address for the virtual router with VRID of 15 on VLAN 20

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 15 ip 10.2.4.5 secondary
switch(config-if-vl20)#
```

vrrp preempt

The **vrrp preempt** command controls a virtual router's preempt mode setting. When preempt mode is enabled, the switch assumes the role of master virtual router if it has a higher priority than the current master router. When preempt mode is disabled, the switch can become the master virtual router only when a master virtual router is not present on the subnet, regardless of vrrp priority settings. By default, preempt mode is enabled.

The **no vrrp preempt** and **default vrrp preempt** commands disable preempt mode for the specified virtual router; the **default vrrp prempt** command stores a corresponding **no vrrp preempt** statement in *running-config*. The **vrrp preempt** command enables preempt mode by removing the corresponding **no vrrp preempt** statement from *running-config*.

The **no vrrp** command also enables preempt mode by removing the **no vrrp preempt** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group preempt
no vrrp group preempt
default vrrp group preempt
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.

Related Commands

vrrp preempt delay

Examples

- This command disables preempt mode for virtual router 20 on VLAN 40.

```
switch(config)#interface vlan 40
switch(config-if-vl40)#no vrrp 20 preempt
switch(config-if-vl40)#
```

- This command enables preempt mode for virtual router 20 on VLAN 40.

```
switch(config-if-vl40)#vrrp 20 preempt
switch(config-if-vl40)#
```

vrrp preempt delay

The **vrrp preempt delay** command specifies the interval between a VRRP preemption event and the point when the switch becomes the master vrrp router. A preemption event is any event that results in the switch having the highest virtual router priority setting while preemption is enabled. The **vrrp preempt** command enables preemption for a specified virtual router.

The command configures two delay periods:

- **minimum** time delays master vrrp takeover when VRRP is fully implemented.
- **reload** time delays master vrrp takeover after VRRP is initialized following a switch reload (boot). The switch bypasses the reload time to become the VRRP master immediately if it senses there are no other active switches in the virtual router group.

Running-config maintains separate delay statements for **minimum** and **reload** parameters. Commands may list both parameters. Commands that list one parameter do not affect the omitted parameter. Values range from 0 to 3600 seconds (one hour). The default delay is zero seconds for both parameters.

The **no vrrp preempt delay** and **default vrrp preempt delay** commands reset the specified delay to the default of zero seconds. Commands that do no list either parameter resets both periods to zero. The **no vrrp** command also removes all **vrrp preempt delay** commands for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group preempt delay [MINIMUM_DELAY] [RELOAD_DELAY]
no vrrp group preempt delay [minimum] [reload]
default vrrp group preempt delay [DELAY_TYPE]
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- ***MINIMUM_DELAY*** period between preempt event and takeover of master vrrp router role.
  - <no parameter> minimum delay is not altered by command.
  - **minimum** *min_time* delay during normal operation (seconds). Values range from 0 to 3600.
- ***RELOAD_DELAY*** period after reboot-VRRP initialization and takeover of master vrrp router role.
  - <no parameter> reload delay is not altered by command.
  - **reload** *reload_time* delay after reboot (seconds). Values range from 0 to 3600.
- ***DELAY_TYPE*** delay type reset to default by **no** and **default vrrp preempt delay** commands.
  - <no parameter> reload and minimum delays are reset to default.
  - **minimum** minimum delay is reset to default.
  - **reload** reload delay are is to default.

Related Commands

vrrp preempt

Examples

- This command sets the minimum preempt time of 90 seconds for virtual router 20 on VLAN 40.

```
switch(config)#interface vlan 40
switch(config-if-vl40)#vrrp 20 preempt delay minimum 90
switch(config-if-vl40)#
```

- This command sets the minimum and reload preempt time to zero for virtual router 20 on VLAN 40.

```
switch(config-if-vl40)#no vrrp 20 preempt delay
switch(config-if-vl40)#
```

vrrp priority

The **vrrp priority** command configures the switch's priority setting for a VRRP virtual router. Priority values range from 1 to 254. The default value is 100.

The router with the highest vrrp priority setting for a group becomes the master virtual router for that group. The master virtual router controls the IP address of the virtual router and is responsible for forwarding traffic sent to this address. The **vrrp preempt** command controls the time when a switch can become the master virtual router.

The **no vrrp priority** and **default vrrp priority** commands restore the default priority of 100 to the virtual router on the configuration mode interface by removing the corresponding **vrrp priority** command from ***running-config***. The **no vrrp** command also removes the **vrrp priority** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

vrrp *group* **priority** *level*
no vrrp *group* **priority**
default vrrp *group* **priority**

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *level* priority setting for the specified virtual router. Values range from 1 to 254.

Examples

- This command sets the virtual router priority value of 250 for virtual router group 45 on VLAN 20.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 45 priority 250
switch(config-if-vl20)#
```

vrrp shutdown

The **vrrp shutdown** command places the switch in stopped state for the specified virtual router. While in stopped state, the switch cannot act as a Master or backup router for the virtual router group.

The **no vrrp shutdown** and **default vrrp shutdown** commands remove the corresponding **vrrp shutdown** command from ***running-config***. This changes the switch's virtual router state to ***backup*** or ***master*** if the virtual router is properly configured.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

vrrp *group* **shutdown**
no vrrp *group* **shutdown**
default vrrp *group* **shutdown**

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.

Example

- This command places the switch in stopped mode for virtual router 24 on VLAN 20.

```
switch(config)#interface vlan 20
switch(config-if-vl20)#vrrp 24 shutdown
switch(config-if-vl20)#
```

- This command moves the switch out of stopped mode virtual router 24 on VLAN 20.

```
switch(config-if-vl20)#no vrrp 24 shutdown
switch(config-if-vl20)#
```

vrrp timers advertise

The **vrrp timers advertise** command configures the interval between successive advertisement messages that the switch sends to VRRP routers in the specified virtual router group. The switch must be the group's Master virtual router to send advertisement messages. The advertisement interval must be configured identically on all physical routers in the virtual router group.

The advertisement interval also influences the timeout interval that defines when the virtual router becomes the master virtual router. When preemption is enabled, the virtual router becomes the master when three times the advertisement interval elapses after the switch detects master router priority conditions.

The **no vrrp timers advertise** and **default vrrp timers advertise** commands restore the default advertisement interval of one second for the specified virtual router by removing the corresponding **vrrp timers advertise** command from ***running-config***. The **no vrrp** command also removes the **vrrp timers advertise** command for the specified virtual router.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group timers advertise adv_time
no vrrp group timers advertise
default vrrp group timers advertise
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *adv_time* advertisement interval (seconds). Values range from 1 to 255. Default value is 1.

Example

- This command sets the advertisement interval of five seconds for the virtual router 35 on VLAN 100.

```
switch(config)#interface vlan 100
switch(config-if-vl100)#vrrp 35 timers advertise 5
switch(config-if-vl100)#
```

vrrp track

The **vrrp track** command configures the VRRP client process on the configuration mode interface to track the specified tracked object and react to changes in its status.

The **no track** and **default track** commands remove the specified tracked object by removing the corresponding **track** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
vrrp group track object_name ACTION
no vrrp group track object_name ACTION
default vrrp group track object_name ACTION
```

Parameters

- *group* virtual router identifier (VRID). Values range from 1 to 255.
- *object_name* name of tracked object.
- ***ACTION*** The action that VRRP is to take when the tracked object changes status. Options include:
  - **decrement** *amount* decrease VRRP priority by *amount*.
  - **shutdown** shut down VRRP on the configuration mode interface.

Example

- This command causes Ethernet interface 5 to disable VRRP when tracked object ETH8 changes state.

```
switch(config-if-Et5)#vrrp 1 track ETH8 shutdown
switch(config-if-Et5)#
```

Chapter 15

Spanning Tree Protocol

Spanning Tree Protocols prevent bridging loops in Layer 2 Ethernet networks. Arista switches support Rapid Spanning Tree, Multiple Spanning Tree, and Rapid-Per VLAN Spanning Tree protocols.

These sections describe the Arista Spanning Tree Protocol implementation.

- Section 15.1: Introduction to Spanning Tree Protocols
- Section 15.2: Spanning Tree Overview
- Section 15.3: Configuring a Spanning Tree
- Section 15.4: STP Commands

15.1 Introduction to Spanning Tree Protocols

Arista Switches support the leading spanning tree protocols: RSTP, MST and Rapid-PVST. This variety of options simplifies integration into existing networks without compromising network reliability, scalability or performance.

15.2 Spanning Tree Overview

An Ethernet network functions properly when only one active path exists between any two stations. A spanning tree is a loop-free subset of a network topology. Spanning Tree Protocol (STP) is a Layer 2 network protocol that ensures a loop-free topology for any bridged Ethernet LAN. STP allows a network to include spare links as automatic backup paths that are available when an active link fails without creating loops or requiring manual intervention. The original STP is standardized as IEEE 802.1D.

Several variations to the original STP improve performance and add capacity. Arista switches support these STP versions:

- Rapid Spanning Tree (RSTP)
- Multiple Spanning Tree (MSTP)
- Rapid Per-VLAN Spanning Tree (Rapid-PVST)

The Overview contains the following sections:

- Section 15.2.1: Spanning Tree Protocol Versions
- Section 15.2.2: Structure of a Spanning Tree Instance
- Section 15.2.3: BPDUs

15.2.1 Spanning Tree Protocol Versions

STP versions supported by Arista switches address two limitations of the original Spanning Tree protocol that was standardized as IEEE 802.1D:

- Slow convergence to the new spanning tree topology after a network change
- The entire network is covered by one spanning tree instance.

The following sections describe the supported STP versions, compatibility issues in networks containing switches running different STP versions, and supported alternatives to spanning tree.

15.2.1.1 Rapid Spanning Tree Protocol (RSTP)

RSTP is specified in 802.1w and supersedes STP. RSTP provides rapid convergence after network topology changes. RSTP provides a single spanning tree instance for the entire network, similar to STP. Standard 802.1D-2004 incorporates RSTP and obsoletes STP.

The RSTP instance the base unit of MST and Rapid-PVST spanning trees.

15.2.1.2 Rapid Per-VLAN Spanning Tree Protocol (Rapid-PVST)

Per-VLAN Spanning Tree (PVST) extends the original STP to support a spanning tree instance on each VLAN in the network. The quantity of PVST instances in a network equals the number of configured VLANs, up to a maximum of 4094 instances. PVST can load balance layer-2 traffic without creating a loop because it handles each VLAN as a separate network. However, PVST does not address slow network convergence after a network topology change.

Arista switches support Rapid-PVST, which is a variation of PVST based on RSTP instances. Rapid-PVST provides rapid connectivity recovery after the failure of a bridge, port, or LAN. Rapid-PVST can be enabled or disabled on individual VLANs.

15.2.1.3 Multiple Spanning Tree Protocol (MSTP)

MST extends RSTP to support multiple spanning tree instances on a network. This extension provides both rapid convergence and load balancing in a VLAN environment. MST is backward compatible with Rapid Spanning Tree Protocol (RSTP). By default, Arista switches use MSTP.

MST supports multiple spanning tree instances, similar to Rapid PVST. However, MST associates an instance with multiple VLANs. This architecture supports load balancing by providing multiple forwarding paths for data traffic. Network fault tolerance is improved because failures in one instance do not affect other instances.

MST Regions

An *MST region* is a set of interconnected bridges with the same MST configuration. Each region can support a maximum of 65 spanning-tree instances. MST regions are identified by a version number, name, and VLAN-to-instance map; these parameters must be configured identically on all switches in the region. Only MST region members participate with the MST instances defined in the region. A VLAN can be assigned to only one spanning-tree instance at a time. MST does not specify the maximum number of regions that a network can contain.

MST Instances

Each MST instance is identified by an instance number that ranges from 0 to 4094 and is associated with a set of VLANs. An MST region contains two types of spanning tree instances: an internal spanning tree instance (IST) and multiple spanning tree instances (MSTI).

- The *Internal Spanning Tree Instance* (IST) is the default spanning tree instance in an MST region and is always instance 0. It provides the root switch for the region and contains all VLANs configured on the switch that are not assigned to a MST instance.
- *Multiple Spanning Tree instances* (MSTI) consists of VLANs that are assigned through MST configuration statements. VLANs assigned to an MSTI are removed from the IST instance. VLANs in an MSTI operate as a part of a single Spanning Tree topology. Because each VLAN can belong to only one instance, MST instances (and the IST) are topologically independent.

15.2.1.4 Version Interoperability

A network can contain switches running different spanning tree versions. The common spanning tree (CST) is a single forwarding path the switch calculates for STP, RSTP, MSTP, and Rapid-PVST topologies in networks containing multiple spanning tree variations.

In multi-instance topologies, the following instances correspond to the CST:

- **Rapid-PVST**: VLAN 1
- **MST**: IST (instance 0)

RSTP and MSTP are compatible with other spanning tree versions:

- An RSTP bridge sends 802.1D (original STP) BPDUs on ports connected to an STP bridge.
- RSTP bridges operating in 802.1D mode remain in 802.1D mode even after all STP bridges are removed from their links.
- An MST bridge can detect that a port is at a region boundary when it receives an STP BPDU or an MST BPDU from a different region.
- MST ports assume they are boundary ports when the bridges to which they connect join the same region.

The **clear spanning-tree detected-protocols** command forces MST ports to renegotiate with their neighbors.

RSTP provides backward compatibility with 802.1D bridges as follows:

- RSTP selectively sends 802.1D-configured BPDUs and Topology Change Notification (TCN) BPDUs on a per-port basis.
- When a port initializes, the migration delay timer starts and RSTP BPDUs are transmitted. While the migration delay timer is active, the bridge processes all BPDUs received on that port.
- If the bridge receives an 802.1D BPDU after a port's migration delay timer expires, the bridge assumes it is connected to an 802.1D bridge and starts using only 802.1D BPDUs.
- When RSTP uses 802.1D BPDUs on a port and receives an RSTP BPDU after the migration delay expires, RSTP restarts the migration delay timer and resumes using RSTP BPDUs on that port.

15.2.1.5 Switchport Interface Pairs

Switchport interface pairs associate two interfaces in a primary-backup configuration. When the primary interface is functioning, the backup interface remains dormant in standby mode. When the primary interface stops functioning, the backup interface handles the traffic.

An alternative implementation balances traffic between the primary and backup interfaces. If either interface shuts down, the other handles traffic addressed to the pair.

The following guidelines apply to switchport interface pairs.

- Ethernet and Port Channels can be primary interfaces.

- Ethernet, Port Channel, Management, Loopback, and VLAN interfaces can be backup interfaces.
- The primary and backup interfaces can be different interface types.
- Interface pairs should be similarly configured to ensure consistent behavior.
- An interface can be associated with a maximum of one backup interface.
- An interface can back up a maximum of one interface.
- Any Ethernet interface configured in an interface pair cannot be a port channel member.
- STP is disabled on ports configured as primary or backup interfaces.
- Static MAC addresses should be configured after primary-backup pairs are established.

15.2.1.6 Disabling Spanning Tree

When spanning tree is disabled and switchport interface pairs are not configured, all interfaces forward packets as specified by their configuration. STP packets are not generated and inbound STP packets are forwarded on the VLAN where they are received as normal multicast data packets.

Important Disabling all Spanning Tree Protocols on the switch is strongly discouraged.

15.2.2 Structure of a Spanning Tree Instance

A layer 2 network consists of bridges and network segments. A loop exists when multiple active paths connect two components. Spanning tree protocols allow only one active path between any two network components. Loops are removed by blocking selected ports that connect bridges to network segments.

Ports are assigned cost values that reflect their transmission speed and any other criteria selected by the administrator. Ports with faster transmission speeds and other desirable characteristics are assigned lower costs. High cost ports are blocked in deference to lower cost ports.

A network topology defines multiple possible spanning trees. Network bridges collectively compute and implement one spanning tree to maintain connectivity between all network components while blocking ports that could result in loops. Administrators improve network performance by adjusting parameter settings to select the most efficient spanning tree.

Spanning tree bridges continuously transmit topology information to notify all other bridges on the network when topology changes are required, such as when a link fails. Bridge Protocol Data Units (BPDUs) are STP information packets that bridges exchange.

The following sections describe spanning tree configuration parameters.

15.2.2.1 Root and Designated Bridges

The **root bridge** is the center of the STP topology. A spanning tree instance has one root bridge. Spanning tree bases path calculations on each network component's distance from the root bridge.

All other network bridges calculate paths to the Root Bridge when selecting spanning tree links. STP calculates the distance to the Root Bridge to build a loop-free topology that features the shortest distance between devices among all possible paths.

Each switch is assigned a unique Bridge ID number for each instance. All network switches collectively elect the Root Bridge by comparing Bridge IDs. The root bridge is the switch with the lowest Bridge ID.

The Bridge ID is contains the following eight bytes, in order of decreasing significance:

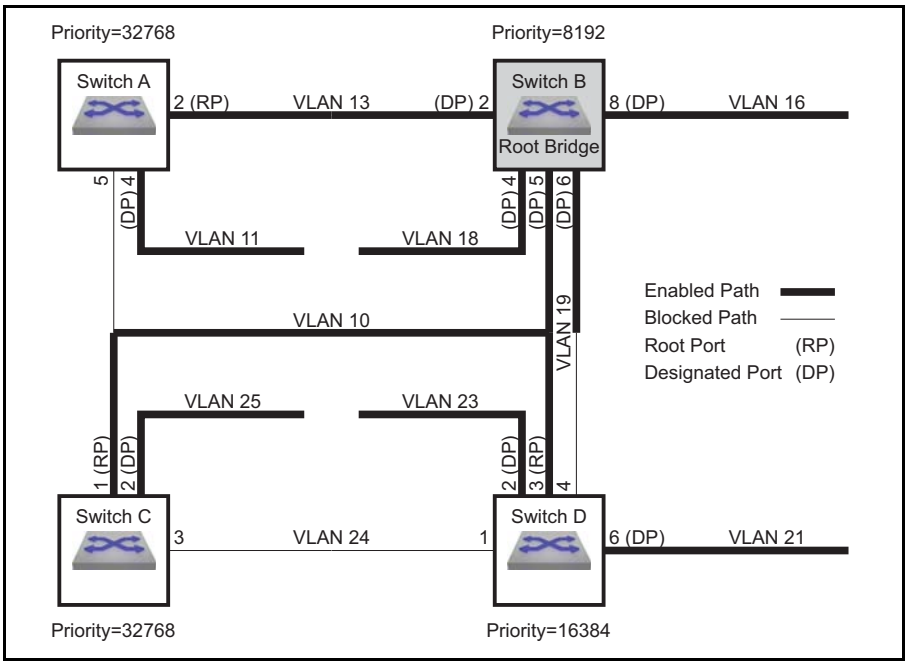
- Port Priority (four bits)
- Instance number (12 bits): VLAN number (Rapid-PVST); Instance number (MST); 0 (RST)
- MAC address of switch (six bytes)

A **designated bridge** is defined for each network segment as the switch that provides the segment's shortest path to the root bridge. A designated bridge is selected for each segment after a root bridge is selected; a switch can be a designated bridge for multiple segments.

The following network calculations in Figure 15-1 assume that each path has the same cost:

- Switch B is the root bridge – its Bridge ID is lowest because it has the smallest port priority.
- Switch A is the designated bridge for VLAN 11.
- Switch B is the designated bridge for VLAN 10, VLAN 13, VLAN 16, VLAN 18, VLAN 19.
- Switch C is the designated bridge for VLAN 25.
- Switch D is the designated bridge for VLAN 21, VLAN 23.

Figure 15-1 Spanning Tree Network Example



15.2.2.2 Port Roles

Messages from any connected device to the root bridge traverse a least-cost path, which has the smallest cost among all possible paths to the root bridge. The cost of a path is the sum of the costs of all path segments, as defined through port cost settings.

Active ports in a least cost-path fulfill one of two possible roles: root port and designated port. STP blocks all other network ports. STP also defines alternate and backup ports to handle traffic when an active port is inaccessible.

- **Root port (RP) accesses the bridge's least-cost path to the root bridge.** Each bridge selects its root port after calculating the cost of each possible path to the root bridge.

  The following ports in Figure 15-1 are root ports:

  Switch A: port 2
  Switch C: port 1
  Switch D: port 3

- **Designated port (DP) accesses a network segment's designated bridge.** Each segment defines one DP. Switches can provide DPs for multiple segments. All ports on the root bridge are DPs.

The following ports in Figure 15-1 are designated ports:

Switch A: port 4 (VLAN 11)
Switch B: port 2 (VLAN 13), port 4 (VLAN 18), port 5 (VLAN 10), port 6 (VLAN 19), port 8 (VLAN 16)
Switch C: port 2 (VLAN 25)
Switch D: port 2 (VLAN 23), port 6 (VLAN 21)

- **Alternate ports** provide backup paths from their bridges to the root bridge. An alternate port is blocked until a network change transforms it into a root port.
- **Backup ports** provide alternative paths from VLANs to their designated bridges. A backup port is blocked until a network change transforms it into a designated port.

15.2.2.3 Port Activity States

A port's activity state defines its current STP activity level. STP monitors BPDUs for network changes that require an activity state transition.

STP defines five port activity states:

- **Forwarding:** The port receives and sends data. Root ports and designated ports are either in, or transitioning to, this state.
- **Blocking:** The port does not receive or send data. Blocked ports receive BPDU packets. All ports except RPs and DPs are blocked, including alternate and backup ports.
- **Listening:** The first transitional post-blocking state, usually resulting from a network change that transforms a port into a root or designated port.
- **Learning:** The last transitional post-blocking state where the port prepares to forward frames by adding source addresses from inbound data packets to the switching database.
- **Disabled:** The interface does not forward frames or receive BPDU packets. Ports are manually disabled and not included in spanning tree calculations or operations.

15.2.2.4 Port Types

Port type is a configurable parameter that reflects the type of network segment that is connected to the port. Proper port type configuration results in rapid convergence after network topology changes. RSTP port types include normal, network, and edge ports. ***Normal*** is the default port type.

- **Normal** ports have an unspecified topology.
- **Network** ports connect only to switches or bridges.

  RSTP immediately transitions network ports to the blocking state.
- **Edge** ports connect directly to end stations.

  Edge ports transition directly to forwarding state, bypassing listening and learning states, because they do not create loops. An edge port becomes a normal port when it receives a BPDU.

15.2.2.5 Link Types

Link type is a configurable parameter that determines candidates for RSTP fast state transition.

- the default link type for full-duplex ports is ***point-to-point***.
- the default link type for half-duplex ports is ***shared***.

Fast state transitions are allowed on point-to-point links that connect bridges. Fast state transitions are not allowed on shared ports regardless of the duplex setting.

15.2.3 BPDUs

Spanning tree rules specify a root bridge, select designated bridges, and assign roles to ports. STP rule implementation requires that network topology information is available to each switch. Switches exchange topology information through Bridge Protocol Data Units (BPDUs). Information provided by BPDU packets include bridge IDs and root path costs.

15.2.3.1 BPDU Types

STP defines three BPDU types:

- Configuration BPDU (CBPDU), used for computing Spanning Tree.
- Topology Change Notification (TCN) BPDU, announces network topology changes.
- Topology Change Notification Acknowledgment (TCA), acknowledges topology changes.

Bridge enter the following addresses in outbound BPDU frames:

- source address: outbound port's MAC address.
- destination address: STP multicast address 01:80:C2:00:00:00.

Bridges regularly exchange BPDUs to track network changes that trigger STP recomputations and port activity state transitions. The *hello timer* specifies the period between consecutive BPDU messages; the default is two seconds.

15.2.3.2 Bridge Timers

Bridge timers specify parameter values that the switch includes in BPDU packets that it sends as a root bridge. Bridge timers include:

- **hello-time**: transmission interval between consecutive BPDU packets.
- **forward-time**: the period that ports remain in listening and learning states.
- **max-age**: the period that BPDU data remains valid after it is received.
- **max-hop**: the number of bridges in an MST region that a BPDU can traverse before it is discarded.

The switch recomputes the spanning tree topology if it does not receive another BPDU before the max-age timer expires. When ***edge*** ports and ***point-to-point*** links are properly configured, RSTP network convergence does not require forward-delay and max-age timers.

15.2.3.3 MSTP BPDUs

MSTP BPDUs are targeted at a single instance and provide STP information for the entire region. MSTP encodes a standard BPDU for the IST, then adds region information and MST instance messages for all configured instances, where each message conveys spanning tree data for an instance. Frames assigned to VLANs operate in the instance to which the VLAN is assigned. Bridges enter an MD5 digest of the VLAN-to-instance map table in BPDUs to avoid including the entire table in each BPDU. Recipients use this digest and other administratively configured values to identify bridges in the same MST region.

MSTP BPDUs are compatible with RSTP. RSTP bridges view an MST region as a single-hop RSTP bridge regardless of the number of bridges inside the region because:

- RSTP bridges interpret MSTP BPDUs as RSTP BPDUs.
- RSTP bridges increment the ***message age timer*** only once while data flows through an MST region; MSTP measures ***time to live*** with a **remaining hops** variable, instead of the **message age timer**.

Ports at the edge of an MST region connecting to a bridge (RSTP or STP) or to an endpoint are *boundary ports*. These ports can be configured as edge ports to facilitate rapid changes to the forwarding state when connected to endpoints.

15.3 Configuring a Spanning Tree

These sections describe the following configuration processes:

- Section 15.3.1: Version Configuration and Instance Creation
- Section 15.3.2: Spanning Tree Instance Configuration
- Section 15.3.3: Port Roles and Rapid Convergence
- Section 15.3.4: Configuring BPDU Transmissions

15.3.1 Version Configuration and Instance Creation

The switch supports three STP versions and switchport backup interface pairs. Disabling spanning tree is also supported but not recommended.

The **spanning-tree mode** global configuration command specifies the spanning tree version the switch runs. This section describes command options that enable and configure STP versions.

15.3.1.1 Multiple Spanning Tree (MST)

Multiple Spanning Tree is enabled by the **spanning-tree mode** command with the ***mstp*** option. MSTP is the default STP version.

Example

- This command enables Multiple Spanning Tree.

```
switch(config)#spanning-tree mode mstp
switch(config)#
```

Configuring MST Regions

All switches in an MST region must have the same name, revision, and VLAN-to-instance map. MST configuration mode commands sets the region parameters. MST configuration mode is a group-change mode where changes are saved by exiting the mode.

Example

- The **spanning-tree mst configuration** command places the switch in MST configuration mode.

```
switch(config)#spanning-tree mst configuration
switch(config-mst)#
```

The **instance** command assigns VLANs to MST instances. The **name (mst-configuration mode)** and **revision (mst-configuration mode)** commands configure the MST region name and revision.

Examples

- These commands assign VLANs 4-7 and 9 to instance 8 and remove VLAN 6 from instance 10.

```
switch(config-mst)#instance 8 vlans 4-7,9
switch(config-mst)#no instance 10 vlans 6
switch(config-mst)#
```

- These commands assign the **name** (*corporate_1*) and **revision** (*3*) to the switch.

```
switch(config-mst)#name corporate_1
switch(config-mst)#revision 3
switch(config-mst)#
```

The **exit (mst-configuration mode)** command transitions the switch out of MST configuration mode and saves all pending changes. The **abort (mst-configuration mode)** command exits MST configuration mode without saving the pending changes.

Example

- This command exits MST configuration mode and saves all pending changes.

```
switch(config-mst)#exit
switch(config)#
```

Configuring MST Instances

These spanning-tree commands provide an optional MST instance parameter. These commands apply to instance 0 when the optional parameter is not included.

- **spanning-tree priority**
- **spanning-tree root**
- **spanning-tree port-priority**

Examples

- This command configures priority for MST instance 4.

```
switch(config)#spanning-tree mode mst 4 priority 4096
switch(config)#
```

- Each of these commands configure priority for MST instance 0.

```
switch(config)#spanning-tree mode mst 0 priority 4096
```

or

```
switch(config)#spanning-tree mode priority 4096
```

15.3.1.2 Rapid Spanning Tree (RST)

Rapid spanning tree is enabled through the **spanning-tree mode** command with the ***rstp*** option.

Example

- This command enables Rapid Spanning Tree.

```
switch(config)#spanning-tree mode rstp
switch(config)#
```

These spanning-tree commands, when they do not include an optional MST or VLAN parameter, apply to RSTP. Commands that configure MSTP instance 0 also apply to the RSTP instance.

- **spanning-tree priority**
- **spanning-tree root**
- **spanning-tree port-priority**

Examples

- These commands apply to the RST instance.

```
switch(config)#spanning-tree priority 4096
```

and

```
switch(config)#spanning-tree mst 0 priority 4096
```

- These commands do not apply to the RST instance.

```
switch(config)#spanning-tree mst 4 priority 4096
```

and

```
switch(config)#spanning-tree VLAN 3 priority 4096
```

Show commands (such as **show spanning-tree**) displays the RSTP instance as MST0 (MST instance 0).

Example

- This command, while the switch is in RST mode, displays RST instance information.

```
switch(config)#show spanning-tree
MST0
  Spanning tree enabled protocol rstp              <---RSTP mode indicator
  Root ID    Priority    32768
             Address     001c.730c.1867
             This bridge is the root

  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     001c.730c.1867
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et51             designated forwarding 2000      128.51   P2p

switch(config)#
```

15.3.1.3 Rapid Per-VLAN Spanning Tree (Rapid-PVST)

Rapid-PVST mode is enabled by the **spanning-tree mode** command with the ***rapid-pvst*** option.

Example

- This command enables Rapid Per-VLAN Spanning Tree.

```
switch(config)#spanning-tree mode rapid-pvst
switch(config)#
```

These commands provide an optional VLAN parameter for configuring Rapid-PVST instances.

- **spanning-tree priority**
- **spanning-tree root**
- **spanning-tree port-priority**

Example

- This command configures bridge priority for VLAN 4.

```
switch(config)#spanning-tree VLAN 4 priority 4096
switch(config)#
```

15.3.1.4 Switchport Backup Mode

Switchport backup interface pairs is enabled through the **spanning-tree mode** command with the ***backup*** option. Enabling switchport backup disables all spanning-tree modes.

Example

- This command enables switchport backup.

```
switch(config)#spanning-tree mode backup
switch(config)#
```

The **switchport backup interface** command establishes an interface pair between the command mode interface (primary) and the interface specified by the command (backup).

Example

- These commands establish Ethernet interface 7 as the backup port for Ethernet interface 1.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#switchport backup interface ethernet 7
switch(config-if-Et1)#
```

The *prefer* option of the **switchport backup interface** command establishes a peer relationship between the primary and backup interfaces and specifies VLAN traffic that the backup interface normally carries. If either interface goes down, the other interface carries traffic normally handled by both interfaces.

Example

These steps perform the following:

- configures Ethernet interface 1 as a trunk port that handles VLANs 4 through 9 traffic.
- configures Ethernet interface 2 as the backup interface.
- assigns Ethernet 2 as the preferred interface for VLANs 7 through 9.

Step 1 Enter configuration mode for the primary interface

```
switch(config)#interface ethernet 1
```

Step 2 Configure the primary interface as a trunk port that services VLANs 4-9

```
switch(config-if-Et1)#switchport mode trunk
switch(config-if-Et1)#switchport trunk allowed vlan 4-9
```

Step 3 Configure the backup interface and specify the VLANs that it normally services.

```
switch(config-if-Et1)#switchport backup Ethernet 2 prefer vlan 7-9
switch(config-if-Et1)#
```

15.3.1.5 Disabling Spanning Tree

Spanning tree is disabled by the **spanning-tree mode** command with the *none* option. The switch does not generate STP packets. Switchport interfaces forward packets when connected to other ports. The switch forwards inbound STP packets as multicast data packets on the VLAN where they are received.

Example

- This command disables all spanning-tree functions.

```
switch(config)#spanning-tree mode none
switch(config)#
```

15.3.2 Spanning Tree Instance Configuration

A network performs these steps to set up an STP instance:

1. The bridge with the lowest ID is elected root bridge.
2. Root ports (RP) are selected on all other bridges.
3. Designated bridges are selected for each network segment.
4. Designated ports (DP) are selected on each designated bridge.
5. Networks begin forwarding data through RPs and DPs. All other ports are blocked.

15.3.2.1 Root Bridge Parameters

STPs use bridge IDs for electing the Root Bridge. Switches denote a Bridge ID for each configured Spanning Tree instance. The bridge ID composition is

- Priority (four bits)

  Priority is expressed as a multiple of 4096 because it is stored as the four most significant bits of a two-byte number.

- Protocol Dependent (twelve bits)

  — Rapid-PVST: VLAN number
  — MST: Instance number
  — RST: 0

- MAC address of switch (six bytes)

Example

This command displays a table of root bridge information.

```
switch>show spanning-tree root
                    Root ID          Root    Hello  Max  Fwd
Instance     Priority    MAC addr    Cost    Time   Age  Dly  Root Port
----------   ------------------- --------- -----  ---  ---  ------------
MST0         32768 001c.7301.23de         0    2    20  15   Po937
MST101       32869 001c.7301.23de      3998    0     0   0   Po909
MST102       32870 001c.7301.23de      3998    0     0   0   Po911
```

The switch defines bridge IDs for three MST instances:

- MST 0: 32768 (Priority (32768)+Instance number(0)) and 001c.7301.23de (MAC address)
- MST101: 32869 (Priority (32768)+Instance number(101)) and 001c.7301.23de (MAC address)
- MST102: 32870 (Priority (32768)+Instance number(102)) and 001c.7301.23de (MAC address)

The switch provides two commands that configure the switch priority: **spanning-tree priority** and **spanning-tree root**. The commands differ in the available parameter options:

- **spanning-tree priority** options are integer multiples of 4096 between 0 and 61440.
- **spanning-tree root** options are ***primary*** and ***secondary***.

  — ***primary*** assigns a priority of 8192.
  — ***secondary*** assigns a priority of 16384.

The default priority value is 32768.

The following examples configure Bridge IDs with both commands.

Example

- These commands configure MST instance bridge priorities with the **root** command:

```
switch(config)#spanning-tree mst 0 root primary
switch(config)#spanning-tree mst 1 root secondary
switch>show spanning-tree root
                    Root ID          Root    Hello  Max  Fwd
Instance     Priority    MAC addr    Cost    Time   Age  Dly  Root Port
----------   ------------------- --------- -----  ---  ---  ------------
MST0          8192 001c.7301.6017         0    2    20  15   None
MST1         16385 001c.7301.6017         0    0     0   0   None
MST2         32770 001c.7301.6017         0    0     0   0   None
```

  — Instance 0 root priority is 8192: primary priority plus the instance number of 0.
  — Instance 1 root priority is 16385: secondary priority plus the instance number of 1.
  — Instance 2 root priority is 32770: default priority plus the instance number of 2.

These priority settings normally program the switch to be the primary root bridge for instance 0, the secondary root bridge for instance 1, and a normal bridge for instance 2.VLAN 4. Primary and secondary root bridge elections also depend on the configuration of other network bridges.

- These prioity commands configure Rapid-PVST VLAN bridge priorities:

```
switch(config)#spanning-tree vlan 1 priority 8192
switch(config)#spanning-tree vlan 2 priority 16384
switch(config)#spanning-tree vlan 3 priority 8192
switch(config)#no spanning-tree vlan 4 priority
switch(config)#show spanning-tree root
                   Root ID          Root    Hello  Max  Fwd
Instance     Priority    MAC addr   Cost    Time   Age  Dly  Root Port
----------   ---------- ---------- ------- -----  ---  ---  ------------
VL1           8193 001c.7301.6017         0    2    20  15   None
VL2          16386 001c.7301.6017         0    2    20  15   None
VL3           8195 001c.7301.6017         0    2    20  15   None
VL4          32788 001c.7301.6017         0    2    20  15   None
```

— VLAN 1 root priority is 8193: configured priority plus the VLAN number of 1.
— VLAN 2 root priority is 16386: configured priority plus the VLAN number of 2.
— VLAN 3 root priority is 8195: configured priority plus the VLAN number of 3.
— VLAN 4 root priority is 32788: default priority plus the VLAN number of 4.

These priority settings normally program the switch to be the primary root bridge for VLANs 1 and 3, the secondary root bridge for VLAN2, and a normal bridge for VLAN 4. Primary and secondary root bridge elections also depend on the configuration of other network bridges.

15.3.2.2 Path Cost

Spanning tree calculates the costs of all possible paths from each component to the root bridge. The path cost is equal to the sum of the cost assigned to each port in the path. Ports are assigned a cost by default or through CLI commands. Cost values range from 1 to 200000000 (200 million).

The default cost is a function of the interface speed:

- 1 gigabit interfaces have a default cost of 20000.
- 10 gigabit interfaces have a default cost of 2000.

The **spanning-tree cost** command configures the path cost of the configuration mode interface. Costs can be specified for Ethernet and port channel interfaces. The command provides a mode parameter for assigning multiple costs to a port for MST instances or Rapid-PVST VLANs.

Examples

- These commands configure a port cost of 25000 to Ethernet interface 5. This cost is valid for RSTP or MSTP instance 0.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree cost 25000
switch(config-if-Et5)#
```

- This command configures a path cost of 300000 to Ethernet interface 5 in MST instance 200.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree mst 200 cost 300000
switch(config-if-Et5)#
```

- This command configures a path cost of 10000 to Ethernet interface 5 in Rapid-PVST VLAN 200-220.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree vlan 200-220 cost 10000
switch(config-if-Et5)#
```

15.3.2.3 Port Priority

Spanning-tree uses the port priority interface parameter to select ports when resolving loops. The port with the lower port priority numerical value is placed in forwarding mode. When multiple ports are assigned equal port priority numbers, the port with the lower interface number is placed in forwarding mode. Valid port-priority numbers are multiples of 16 between 0 and 240; the default is 128.

The **spanning-tree port-priority** command configures the port-priority number for the configuration mode interface. The command provides a mode option for assigning different priority numbers to a port for multiple MST instances or Rapid-PVST VLANs. Port-priority can be specified for Ethernet and port channel interfaces.

Examples

- This command sets the access port priority of 144 for Ethernet 5 interface.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree port-priority 144
switch(config-if-Et5)#
```

- This command sets the access port priority of 144 for Ethernet 5 interface in MST instance 10.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree mst 10 port-priority 144
switch(config-if-Et5)#
```

15.3.3 Port Roles and Rapid Convergence

Spanning Tree provides the following options for controlling port configuration and operation:

- **PortFast**: Allows ports to skip the listening and learning states before entering forwarding state.
- **Port Type** and **Link Type:** Designates ports for rapid transitions to the forwarding state.
- **Root Guard**: Prevents a port from becoming root port or blocked port.
- **Loop Guard**: Prevents loops resulting from a unidirectional link failure on a point-to-point link.
- **Bridge Assurance**: Prevents loops caused by unidirectional links or a malfunctioning switch.

15.3.3.1 PortFast

PortFast is enabled on access ports connected to a single workstation or server to allow those devices immediate network access without waiting for spanning tree convergence. Enabling PortFast on ports connected to another switch can create loops.

A ***portfast*** port that receives a BPDU sets its operating state to ***non-portfast*** while remaining in ***portfast*** configured state. In this state, the port is subject to topology changes and can enter the blocking state.

The **spanning-tree portfast** command programs access ports to immediately enter the forwarding state, bypassing listening and learning states. PortFast connects devices attached to an access port, such as a single workstation, to the network immediately without waiting for STP convergence. PortFast can also be enabled on trunk ports.

Example

- This command unconditionally enables portfast on Ethernet 5 interface.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast
switch(config-if-Et5)#
```

15.3.3.2 Port Type and Link Type Configuration

RSTP only achieves rapid transition to forwarding state on edge ports and point-to-point links.

Port Type

Edge ports are directly connected to end stations. Because edge ports do not create loops, they transition directly to forwarding state, bypassing listening and learning states, when a link is established.

The port type determines the behavior of the port with respect to STP extensions. The **spanning-tree portfast <port type>** command sets the configuration mode interface's port type. Spanning tree ports can be configured as ***edge*** ports, ***network*** ports, or ***normal*** ports. The default port type is ***normal***.

- **Edge ports** connect to a host (end station). Configuring a port that connects to a bridge as an edge port may create a loop. Edge ports that receive a BPDU become a normal spanning tree port.
- **Network ports** connect only to a Layer 2 switch or bridge. Configuring a port connected to a host as a network port transitions the port to the blocking state.
- N**ormal ports** have an unspecified topology.

Example

- This command configures Ethernet 5 interface as a network port.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast network
switch(config-if-Et5)#
```

Auto-edge detection converts ports into edge ports when they do not receive a new BPDU before the current BPDU exires, as measured by the max-age timer. The **spanning-tree portfast auto** command enables auto-edge detection on the configuration mode interface, superseding the **spanning-tree portfast** command. Auto-edge detection is enabled by default.

Example

- This command enables auto-edge detection on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast auto
switch(config-if-Et5)#
```

Link Type

The switch derives a port's default link type from its duplex mode:

- full-duplex ports are ***point-to-point***.
- half-duplex ports are ***shared***.

The **spanning-tree link-type** command specifies the configuration mode interface's link-type. RSTP fast transition is not allowed on ***shared link*** ports, regardless of their duplex setting. Because the ports are full-duplex by default, the default link-type setting is ***point-to-point***.

Example

- This command configures Ethernet 5 interface as a shared port.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree link-type shared
switch(config-if-Et5)#
```

15.3.3.3 Root Guard and Loop Guard

Root guard prevents a port from becoming a root port, which stops connected switches from becoming root bridges. When a switch detects a new root bridge, its root-guard-enabled ports enter blocked (root-inconsistent) state. When the switch no longer detects a new root, these ports enter listening state.

Root guard is enabled on a per-port basis. The setting applies to all STP instances. Disabling root guard places the port in listening state.

The **spanning-tree guard** command, with the root option, enables root guard on the configuration mode interface.

Example

- This command enables root guard on Ethernet 5 interface.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree guard root
switch(config-if-Et5)#
```

Loop guard prevents loops from unidirectional link failures on point-to-point links by verifying that non-designated ports (root, blocked, and alternate) are receiving BPDUs from their designated ports. A loop-guard-enabled root or blocked port that stops receiving BPDUs transitions to the blocking (loop-inconsistent) state. The port recovers from this state when it receives a BPDU.

Loop guard, when enabled globally, applies to all point-to-point ports. Loop guard is configurable on individual ports and applies to all STP instances of an enabled port. Loop-inconsistent ports transition to listening state when loop guard is disabled.

Enabling loop guard on a root switch has no effect until the switch becomes a nonroot switch.

When using loop guard:

- Do not enable loop guard on portfast-enabled ports.
- Loop guard is not functional on ports not connected to point-to-point links.
- Loop guard has no effect on disabled spanning tree instances.

Loop guard aspects on port channels include:

- BPDUs are sent over the channel's first operational port. Loop guard blocks the channel if that link becomes unidirectional even when other channel links function properly.
- Creating a new channel destroys state information for its component ports; new channels with loop-guard-enabled ports can enter forwarding state as a DP.
- Dissembling a channel destroys its state information; component ports from a blocked channel can enter the forwarding state as DPs, even if the channel contained unidirectional links.
- A unidirectional link on any port of a loop-guard-enabled channel blocks the entire channel until the affected port is removed or the link resumes bidirectional operation.

Loop guard configuration commands include:

- **spanning-tree loopguard default** command enables loop guard as a default on all switch ports.
- **spanning-tree guard** control the loop guard setting on the configuration mode interface. This command overrides the default command for the specified interface.

Examples

- This command enables loop guard as the default on all switch ports.

```
switch(config)#spanning-tree loopguard default
switch(config)#
```

- This command enables loop guard on Ethernet 6 interface.

```
switch(config)#interface ethernet 6
switch(config-if-Et6)#spanning-tree guard loop
switch(config-if-Et6)#
```

15.3.3.4 Bridge Assurance

Bridge assurance protects against unidirectional link failures, other software failures, and devices that continue forwarding data traffic after they quit running spanning tree.

Bridge assurance operate only on ***network*** ports with ***point-to-point*** links where bridge assurance is enabled on each side of the link. Bridge assurance-enabled ports are blocked when they link to a port where bridge assurance is not enabled.

Bridge assurance programs the switch to send BPDUs at each hello time period through all bridge assurance enabled ports. Ports not receiving a BPDU packet within an hello time period enter inconsistent (blocking) state and are not used in root port calculations. Blocked ports that begin receiving BPDUs are removed from the inconsistent (blocking) state and resume normal state transitions.

Bridge assurance is enabled by default on all network ports. To disable it, use the **no** form of the **spanning-tree bridge assurance** command. To enable bridge assurance if it is disabled, use the **spanning-tree bridge assurance** command.

Example

- This command enables bridge assurance on the switch.

```
switch(config)#spanning-tree bridge assurance
switch(config)#
```

15.3.4 Configuring BPDU Transmissions

The following sections describe instructions that configure BPDU packet contents and transmissions.

15.3.4.1 Bridge Timers

Bridge timers configure parameter values that the switch includes in BPDU packets that it sends as a root bridge. Bridge timers include:

- **hello-time**: the transmission interval between consecutive outbound BPDU packets.
- **forward-time**: the period that ports are in listening and learning states prior to forwarding packets.
- **max-age**: the period that BPDU data remains valid after it is received. The switch recomputes the spanning tree topology if it does not receive another BPDU packet before the timer expires.
- **max-hop**: the number of bridges in an MST region that a BPDU can traverse before it is discarded.

In standard STP, ports passively wait for **forward_delay** and **max_age** periods before entering the forwarding state. RSTP achieves faster convergence by relying on edge port and link type definitions to start forwarding traffic. When edge ports and link types are properly configured, bridge timers are used in RSTP as backup or when interacting with networks running standard STP.

The **spanning-tree hello-time** command configures the hello time.

Example

- This command configures a hello-time of 1 second (1000 ms).

```
switch(config)#spanning-tree hello-time 1000
switch(config)#
```

The **spanning-tree max-hops** command specifies the max hop setting that the switch inserts into BPDUs that it sends out as the root bridge.

Example

- This command sets the max hop value to 40.

```
switch(config)#spanning-tree max-hops 40
switch(config)#
```

The **spanning-tree forward-time** command configures the forward delay setting that the switch inserts into BPDUs that it sends out as the root bridge.

Example

- This command sets the forward delay timer value to 25 seconds.

```
switch(config)#spanning-tree forward-time 25
switch(config)#
```

The **spanning-tree max-age** command configures the max age setting that the switch inserts into BPDUs that it sends out as the root bridge.

Example

- This command sets the max age timer value to 25 seconds.

```
switch(config)#spanning-tree max-age 25
switch(config)#
```

15.3.4.2 BPDU Transmit Hold-Count

The **spanning-tree transmit hold-count** command specifies the maximum number of BPDUs per second that the switch can send from an interface. Valid settings range from 1 to 10 BPDUs with a default of 6 BPDUs.

Higher hold-count settings can significantly impact CPU utilization, especially in Rapid-PVST mode. Smaller values can slow convergence in some configurations.

Example

- This command configures a transmit hold-count of 8 BPDUs.

```
switch(config)#spanning-tree transmit hold-count 8
switch(config)#
```

15.3.4.3 BPDU Guard

PortFast interfaces do not receive BPDUs in a valid configuration. BPDU Guard provides a secure response to invalid configurations by disabling ports when they receive a BPDU. Disabled ports differ from blocked ports in that they are re-enabled only through manual intervention.

- When configured globally, BPDU Guard is enabled on ports in the operational portfast state.
- When configured on an individual interface, BPDU Guard disables the port when it receives a BPDU, regardless of the port's portfast state.

The **spanning-tree portfast bpduguard default** global configuration command enables BPDU guard by default on all portfast ports. BPDU guard is disabled on all ports by default.

The **spanning-tree bpduguard** interface configuration command controls BPDU guard on the configuration mode interface. This command takes precedence over the default setting configured by **spanning-tree portfast bpduguard default**.

- **spanning-tree bpduguard enable** enables BPDU guard on the interface.
- **spanning-tree bpduguard disable** disables BPDU guard on the interface.
- **no spanning-tree bpduguard** reverts the interface to the default BPDU guard setting.

Example

- These commands enable BPDU guard by default on all portfast ports, then disable BPDU guard on Ethernet 5.

```
switch(config)#spanning-tree portfast bpduguard default
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree bpduguard disable
switch(config-if-Et5)
```

15.3.4.4 BPDU Filter

BPDU filtering prevents the switch from sending or receiving BPDUs on specified ports. BPDU filtering is configurable on Ethernet and port channel interfaces.

Ports with BPDU filtering enabled do not send BPDUs and drops inbound BPDUs. Enabling BPDU filtering on a port not connected to a host can result in loops as the port continues forwarding data while ignoring inbound BPDU packets.

The **spanning-tree bpdufilter** command controls BPDU filtering on the configuration mode interface. BPDU filtering is disabled by default.

Example

- These commands enable BPDU filtering on Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree bpdufilter enable
switch(config-if-Et5)#
```

15.3.4.5 BPDU Rate Limit

BPDU input rate limiting restricts the number of BPDUs that a port with BPDU guard and BPDU filter disabled can process during a specified interval. The port discards all BPDUs that it receives in excess of the specified limit. Configuring the rate limiter requires two steps:

- Establishing the rate limit threshold.
- Enabling rate limiting.

Establishing the Rate Limit Threshold

The **spanning-tree bpduguard rate-limit count (interface)** commands specify BPDU reception rate (quantity per interval) that trigger the discarding of BPDUs. Commands are available in global and interface configuration modes.

- The **spanning-tree bpduguard rate-limit count** global command specifies the maximum reception rate for ports not covered by interface rate limit count commands. The default quantity is 10 times the number of VLANs. The default interval is the hello time (**spanning-tree hello-time**).
- The **spanning-tree bpduguard rate-limit count** interface command defines the maximum BPDU reception rate for the configuration mode interface. The global command specifies the default limit.

Examples

- This command configures the global limit of 5000 BPDUs over a four second interval.

```
switch(config)#spanning-tree bpduguard rate-limit count 5000 interval 4
switch(config)#
```

- These commands configures a limit of 7500 BPDUs over an 8 second interval on Ethernet interface 2.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#spanning-tree bpduguard rate-limit count 7500 interval 8
switch(config-if-Et2)#
```

Enabling Rate Limiting

BPDU rate limiting is enabled globally or on individual ports:

- **spanning-tree bpduguard rate-limit default** enables rate limiting on all ports with no interface rate limiting command. The default setting is ***enabled***.
- **spanning-tree bpduguard rate-limit enable / disable** interface command enables or disables BPDU rate limiting on the configuration mode interface. This command has precedence over the global command.

Examples

- This command enables rate limiting on ports not covered by interface rate limit commands.

```
switch(config)#spanning-tree bpduguard rate-limit default
switch(config)#
```

- These commands enables rate limiting on Ethernet 15.

```
switch(config)#interface ethernet 15
switch(config-if-Et15)#spanning-tree bpduguard rate-limit enable
switch(config-if-Et15)#
```

15.4 STP Commands

Spanning Tree Commands: Global Configuration

- spanning-tree bpduguard rate-limit default . . . . . Page 610
- spanning-tree bpduguard rate-limit count (global) . . . . . Page 608
- spanning-tree bridge assurance . . . . . Page 612
- spanning-tree forward-time . . . . . Page 614
- spanning-tree hello-time . . . . . Page 616
- spanning-tree loopguard default . . . . . Page 618
- spanning-tree max-age . . . . . Page 619
- spanning-tree max-hops . . . . . Page 620
- spanning-tree mode . . . . . Page 621
- spanning-tree mst configuration . . . . . Page 622
- spanning-tree portchannel guard misconfig . . . . . Page 623
- spanning-tree portfast bpdufilter default . . . . . Page 626
- spanning-tree portfast bpduguard default . . . . . Page 627
- spanning-tree priority . . . . . Page 630
- spanning-tree root . . . . . Page 631
- spanning-tree transmit hold-count . . . . . Page 632
- spanning-tree vlan . . . . . Page 633

Spanning Tree Commands: Interface Configuration Mode

- spanning-tree bpduguard rate-limit count (interface) . . . . . Page 609
- spanning-tree bpduguard rate-limit enable / disable . . . . . Page 611
- spanning-tree bpdufilter . . . . . Page 606
- spanning-tree bpduguard . . . . . Page 607
- spanning-tree cost . . . . . Page 613
- spanning-tree guard . . . . . Page 615
- spanning-tree link-type . . . . . Page 617
- spanning-tree port-priority . . . . . Page 629
- spanning-tree portfast . . . . . Page 624
- spanning-tree portfast auto . . . . . Page 625
- spanning-tree portfast <port type> . . . . . Page 628
- switchport backup interface . . . . . Page 634

MST Configuration Commands

- abort (mst-configuration mode) . . . . . Page 583
- exit (mst-configuration mode) . . . . . Page 587
- instance . . . . . Page 588
- name (mst-configuration mode) . . . . . Page 589
- revision (mst-configuration mode) . . . . . Page 590
- show (mst-configuration mode) . . . . . Page 591

Display Commands

- show spanning-tree . . . . . Page 592
- show spanning-tree blockedports . . . . . Page 595
- show spanning-tree bridge . . . . . Page 596
- show spanning-tree counters . . . . . Page 597
- show spanning-tree interface . . . . . Page 598
- show spanning-tree mst . . . . . Page 599
- show spanning-tree mst configuration . . . . . Page 601
- show spanning-tree mst interface . . . . . Page 602
- show spanning-tree mst test information . . . . . Page 603

- show spanning-tree root . . . . . Page 604
- show spanning-tree topology status . . . . . Page 605

Clear Commands

- clear spanning-tree counters . . . . . Page 584
- clear spanning-tree counters session . . . . . Page 585
- clear spanning-tree detected-protocols . . . . . Page 586

abort (mst-configuration mode)

The **abort** command, in MST-Configuration mode, discards pending changes to the MST region configuration, then returns the switch to Global Configuration mode.

The **exit (mst-configuration mode)** command saves MST region changes to ***running-config*** before returning the switch to Global Configuration mode.

Command Mode

MST-Configuration

Command Syntax

abort

Examples

- This command discards changes to the MST region, then returns the switch to Global Configuration mode.

```
Switch(config-mst)#abort
Switch(config)#
```

clear spanning-tree counters

The **clear spanning-tree counters** command resets the BPDU counters for the specified interfaces to zero in all CLI sessions.

Command Mode

Privileged EXEC

Command Syntax

clear spanning-tree counters [***INT_NAME***]

Parameters

- ***INT_NAME*** Interface type and number. Options include:
  - <no parameter> resets counters for all interfaces.
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface management** *m_num* Management interface specified by *m_num*.
  - **interface port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- This command resets the BPDU counters on Ethernet 15 interface.

```
switch#show spanning-tree counters
            Port        Sent      Received       Tagged Error    Other Error
-------------------------------------------------------------------------------
      Ethernet15       32721             0                  0              0
  Port-Channel10        8487             0                  0              0

switch#clear spanning-tree counters interface ethernet 15     <---Clear command
switch#show spanning-tree counters
            Port       Sent       Received       Tagged Error    Other Error
-------------------------------------------------------------------------------
      Ethernet15         11              0                  0              0
  Port-Channel10          8494           2                  6              0

switch#
```

clear spanning-tree counters session

The **clear spanning-tree counter session** command resets the BPDU counters to zero on all interfaces in the current CLI session. Counters in other CLI sessions are not affected.

Command Mode

Privileged EXEC

Command Syntax

clear spanning-tree counters session

Examples

- This command resets the BPDU counters in the current CLI session.

```
switch#show spanning-tree counters
            Port       Sent      Received       Tagged Error    Other Error
-------------------------------------------------------------------------------
      Ethernet15      32721             0                  0              0
  Port-Channel10       8487             0                  0              0

switch#clear spanning-tree counters session
switch#show spanning-tree counters
            Port      Sent       Received       Tagged Error    Other Error
-------------------------------------------------------------------------------
      Ethernet15        11             0                  0              0
  Port-Channel10         7             2                  6              0

switch#
```

clear spanning-tree detected-protocols

The **clear spanning-tree detected-protocols** command restarts the spanning tree protocol (STP) migration state machine on the specified interfaces. The switch is reset to running rapid spanning tree protocol on an interface where it previously detected a bridge running an old version of the protocol.

Command Mode

Privileged EXEC

Command Syntax

```
clear spanning-tree detected-protocols [INT_NAME]
```

Parameters

- *INT_NAME* Interface type and number. Values include
  - <no parameter> all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- This command restarts the STP migration machine on all switch interfaces.

```
switch#clear spanning-tree detected-protocols
switch#
```

exit (mst-configuration mode)

The **exit** command, in MST-Configuration mode, saves changes to the MST region configuration, then returns the switch to Global Configuration mode. MST region configuration changes are also saved by entering a different configuration mode.

Command Mode

MST-Configuration

Command Syntax

```
exit
```

Examples

- This command saves changes to the MST region, then returns the switch to Global Configuration mode.

```
Switch(config-mst)#exit
Switch(config)#
```

- This command saves changes to the MST region, then places the switch in Interface-Ethernet mode.

```
Switch(config-mst)#interface ethernet 3
Switch(config-if-Et3)#
```

instance

The **instance** command inserts an entry into the VLAN-to-instance map that associates a set of VLANs to an MST instance. In addition to defining the MST topology, the VLAN-to-instance map is one of three parameters, along with the MST name and revision number, that identifies the switch's MST region.

The **no instance** command removes specified entries from the VLAN-to-instance map. If the command does not provide a VLAN list, all entries are removed for the specified instance. The **no instance** and **default instance** commands function identically.

Command Mode

MST-Configuration

Command Syntax

```
instance mst_inst vlans v_range
no instance mst_inst [vlans v_range]
no default instance mst_inst [vlans v_range]
```

Parameters

- *mst_inst* MST instance number. Value of *mst_inst* ranges from 0 to 4094.
- *v_range* VLAN list. Formats include a number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command maps VLANs 20-39 to MST instance 2

```
switch(config)#spanning-tree mst configuration
switch(config-mst)#instance 2 vlans 20-39
switch(config-mst)#
```

- This command removes all VLAN mappings to MST instance 10.

```
switch(config-mst)#no instance 10
switch(config-mst)#
```

name (mst-configuration mode)

The **name** command configures the MST region name. The name is one of three parameters, along with the MST revision number and VLAN-to-instance map, that identifies the switch's MST region.

The name consists of up to 32 characters. The default name is an empty string. The name string accepts all characters except the space.

The **no name** and **default name** commands restore the default name by removing the **name** command from ***running-config***.

Command Mode

MST-Configuration

Command Syntax

name *label_text*
no name
default name

Parameters

- *label_text* character string assigned to name attribute. Maximum 32 characters. The space character is not permitted in the name string.

Example

- This command assigns corporate_100 as the MST region name.

```
switch(config)#spanning-tree mst configuration
switch(config-mst)#name corporate_100
switch(config-mst)#show pending
Active MST configuration
Name      [corporate_100]
Revision  0    Instances configured 1

Instance  Vlans mapped
--------  ------------------------------------------------------------------------
0         1-4094
--------------------------------------------------------------------------------
```

revision (mst-configuration mode)

The **revision** command configures the MST revision number. The revision number is one of three parameters, along with the MST name and VLAN-to-instance map, that identifies the switch's MST region. Revision numbers range from 0 to 65535. The default revision number is 0.

The **no revision** and **default revision** commands restore the revision number to its default value by removing the revision command from ***running-config***.

Command Mode

MST-Configuration

Command Syntax

revision *rev_number*
no revision
default revision

Parameters

- *rev_number* revision number. Ranges from 0 to 65535 with a default of 0.

Examples

- This command sets the revision number to 15.

```
switch(config)#spanning-tree mst configuration
switch(config-mst)#revision 15
switch(config-mst)#show pending
Active MST configuration
Name      []
Revision 15   Instances configured 1

Instance  Vlans mapped
--------  ---------------------------------------------------------------
0         1-4094
---------------------------------------------------------------------------
```

show (mst-configuration mode)

The **show** command displays the current and pending MST configuration:

Exiting MST configuration mode stores all pending configuration changes to ***running-config***.

Command Mode

MST-Configuration

Command Syntax

show [***EDIT_VERSION***]

Parameters

- ***EDIT_VERSION*** specifies configuration version that the command displays. Options include:
  - — <no parameter> command displays pending MST configuration.
  - — **active** command displays MST configuration stored in ***running-config***.
  - — **current** command displays MST configuration stored in ***running-config***.
  - — **pending** command displays pending MST configuration.

Example

- These commands contrast the difference between the active and pending configuration by adding MST configuration commands, then showing the configurations.

```
switch(config-mst)#show pending
Active MST configuration
Name      []
Revision  0    Instances configured 1

Instance  Vlans mapped
--------  ----------------------------------------------------------------------
0         1-4094
--------------------------------------------------------------------------------
switch(config-mst)#instance 2 vlan 20-29,102
switch(config-mst)#revision 2
switch(config-mst)#name baseline
switch(config-mst)#show pending
Pending MST configuration
Name      [baseline]
Revision  2    Instances configured 2

Instance  Vlans mapped
--------  ----------------------------------------------------------------------
0         1-19,30-101,103-4094
2         20-29,102
--------------------------------------------------------------------------------
switch(config-mst)#show active

Active MST configuration
Name      []
Revision  0    Instances configured 1

Instance  Vlans mapped
--------  ----------------------------------------------------------------------
0         1-4094
--------------------------------------------------------------------------------
```

show spanning-tree

The **show spanning-tree command** displays spanning tree protocol (STP) data, organized by instance.

Command Mode

EXEC

Command Syntax

```
show spanning-tree [VLAN_ID] [INFO_LEVEL]
```

Parameters

- ***VLAN_ID*** specifies VLANs for which command displays information. Formats include:
  - <no parameter> displays information for all instances VLANs.
  - **vlan** displays data for instances containing the first VLAN listed in ***running-config***.
  - **vlan** *v_range* displays data for instances containing a VLAN in the specified range.
- ***INFO_LEVEL*** specifies level of information detail provided by the command.
  - <no parameter> displays table for each instance listing status, configuration, and history.
  - **detail** displays data blocks for each instance and all ports on each instance.

Display Values

- **Root ID** Displays information on the ROOT ID (elected spanning tree root bridge ID):
  - ***Priority***: Priority of the bridge. Default value is 32768.
  - ***Address***: MAC address of the bridge.
- **Bridge ID** bridge status and configuration information for the locally configured bridge:
  - ***Priority*** Priority of the bridge. The default priority is 32768.
  - ***Address*** MAC address of the bridge.
  - ***Hello Time*** Interval (seconds) between bridge protocol data units (BPDUs) transmissions.
  - ***Max Age*** Maximum time that a BPDU is saved.
  - ***Forward Delay*** Time (in seconds) that is spent in the listening and learning state.
- **Interface** STP configuration participants. Link-down interfaces are not shown.
- **Role** Role of the port as one of the following:
  - ***Root*** The best port for a bridge to a root bridge used for forwarding.
  - ***Designated*** A forwarding port for a LAN segment.
  - ***Alternate*** A port acting as an alternate path to the root bridge.
  - ***Backup*** A port acting as a redundant path to another bridge port.
  - ***Disabled*** A port manually disabled by an administrator.
- **State** Displays the interface STP state as one of the following:
  - ***Listening***
  - ***Learning***
  - ***Blocking***
  - ***Forwarding***
- **Cost** STP port path cost value.
- **Prio. Nbr.** STP port priority. Values range from 0 to 240. Default is 128.
- **Type** The link type of the interface (automatically derived from the duplex mode of an interface):
  - ***P2p Peer (STP)*** Point to point full duplex port running standard STP.
  - ***shr Peer (STP)*** Shared half duplex port running standard STP.

Examples

- This command displays STP data, including a table of port parameters.

```
switch>show spanning-tree vlan 1000
MST0
  Spanning tree enabled protocol rstp
  Root ID    Priority    32768
             Address     001c.7301.07b9
             Cost        1999 (Ext) 0 (Int)
             Port        101 (Port-Channel2)
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     001c.7304.195b
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et4              designated forwarding 20000     128.4    P2p
Et5              designated forwarding 20000     128.5    P2p
Et6              designated forwarding 20000     128.6    P2p
Et23             designated forwarding 20000     128.23   P2p
Et26             designated forwarding 20000     128.26   P2p
Et32             designated forwarding 2000      128.32   P2p

switch>
```

- This command displays output from the show spanning-tree command:

```
Switch#show spanning-tree
MST0
  Spanning tree enabled protocol mstp
  Root ID    Priority    32768
             Address     0011.2201.0301
             This bridge is the root

  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     0011.2201.0301
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et4              designated forwarding 2000      128.4    P2p
Et5              designated forwarding 2000      128.5    P2p
...
PEt4             designated forwarding 2000      128.31   P2p
PEt5             designated forwarding 2000      128.44   P2p
...
Po3               designated forwarding 1999      128.1003 P2p
```

- This command displays STP data, including an information block for each interface running STP.

```
switch>show spanning-tree vlan 1000 detail
 MST0 is executing the rstp Spanning Tree protocol
  Bridge Identifier has priority 32768, sysid 0, address 001c.7304.195b
  Configured hello time 2.000, max age 20, forward delay 15, transmit hold-count 6
  Current root has priority 32768, address 001c.7301.07b9
  Root port is 101 (Port-Channel2), cost of root path is  1999 (Ext) 0 (Int)
  Number of topology changes 4109 last change occurred 1292651 seconds ago
          from Ethernet13

 Port 4 (Ethernet4) of MST0 is designated forwarding
   Port path cost 20000, Port priority 128, Port Identifier 128.4.
   Designated root has priority 32768, address 001c.7301.07b9
   Designated bridge has priority 32768, address 001c.7304.195b
   Designated port id is 128.4, designated path cost 1999 (Ext) 0 (Int)
   Timers: message age 1, forward delay 15, hold 20
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default, Internal
   BPDU: sent 452252, received 0, taggedErr 0, otherErr 0, rateLimiterCount 0
   Rate-Limiter: enabled, Window: 10 sec, Max-BPDU: 400

Port 5 (Ethernet5) of MST0 is designated forwarding
   Port path cost 20000, Port priority 128, Port Identifier 128.5.
   Designated root has priority 32768, address 001c.7301.07b9
   Designated bridge has priority 32768, address 001c.7304.195b
   Designated port id is 128.5, designated path cost 1999 (Ext) 0 (Int)
   Timers: message age 1, forward delay 15, hold 20
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default, Internal
   BPDU: sent 1006266, received 0, taggedErr 0, otherErr 0, rateLimiterCount 0
   Rate-Limiter: enabled, Window: 10 sec, Max-BPDU: 400

<-------OUTPUT OMITTED FROM EXAMPLE-------->

switch>
```

show spanning-tree blockedports

The **show spanning-tree blockedports** command displays the list of blocked (discarding) ports.

Command Mode

EXEC

Command Syntax

show spanning-tree blockedports

Example

- This command displays the ports that are in blocking (discarding) state.

```
switch>show spanning-tree blockedports
Name       Blocked Interfaces List
---------- ------------------------------------------------------------------------
MST0      Po903, Po905, Po907, Po909, Po911, Po913, Po915, Po917, Po919, Po921, Po923

          Po925, Po927, Po929, Po931, Po933, Po935, Po939, Po941, Po943, Po945, Po947


Number of blocked ports (segments) in the system : 22
switch>
```

show spanning-tree bridge

The **show spanning-tree bridge** command displays spanning tree protocol bridge configuration settings for each instance on the switch. The display includes Bridge ID, Hello Time, Max Age, and Forward Delay times.

The command also displays the restartability of the STP agent when the **detail** option is selected. A switch can continuing support of MLAG operation when its peer is offline when the STP agent is unavailable.

Command Mode

EXEC

Command Syntax

show spanning-tree bridge [***INFO_LEVEL***]

Parameters

- *INFO_LEVEL* specifies level of information detail provided by the command.
  - — <no parameter> command displays information in a data table.
  - — **detail** command displays bridge information in data blocks for each instance.

Examples

- This command displays a bridge data table.

```
switch>show spanning-tree bridge
                            Bridge ID                      Hello  Max  Fwd
Instance             Priority             MAC addr         Time   Age  Dly
----------   ---------------------------------------      -----  ---  ---
MST0         32768(32768, sys-id 0   ) 001c.7302.2f98      2000    20   15
MST101       32869(32768, sys-id 101 ) 001c.7302.2f98      2000    20   15
MST102       32870(32768, sys-id 102 ) 001c.7302.2f98      2000    20   15

switch>
```

- This command displays bridge data blocks.

```
switch>show spanning-tree bridge detail
Stp agent is restartable
MST0
  Bridge ID  Priority    32768  (priority 32768 sys-id-ext 0)
             Address     001c.7302.2f98
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec
MST101
  Bridge ID  Priority    32869  (priority 32768 sys-id-ext 101)
             Address     001c.7302.2f98
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec
MST102
  Bridge ID  Priority    32870  (priority 32768 sys-id-ext 102)
             Address     001c.7302.2f98
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec
switch>
```

show spanning-tree counters

The **show spanning-tree counters** command displays the number of BPDU transactions on each interface running spanning tree.

Command Mode

EXEC

Command Syntax

show spanning-tree counters

Example

- This command displays the BPDU counter status on each interface running spanning tree.

```
switch>show spanning-tree counters
         Port      Sent   Received   Tagged Error   Other Error  sinceTimer
-------------------------------------------------------------------------------
    Ethernet2   1008399          0              0             0           0
    Ethernet3   1008554          0              0             0           0
    Ethernet4    454542          0              0             0           0
    Ethernet5   1008556          0              0             0           0
    Ethernet6    827133          0              0             0           0
    Ethernet8   1008566          0              0             0           0
   Ethernet10    390732          0              0             0           0
   Ethernet11   1008559          0              0             0           0
   Ethernet15    391379          0              0             0           0
   Ethernet17    621253          0              0             0           0
   Ethernet19    330855          0              0             0           0
   Ethernet23    245243          0              0             0           0
   Ethernet25    591695          0              0             0           0
   Ethernet26   1007903          0              0             0           0
   Ethernet32   1010429          8              0             0           0
   Ethernet33    510227          0              0             0           0
   Ethernet34    827136          0              0             0           0
   Ethernet38   1008397          0              0             0           0
   Ethernet39   1008564          0              0             0           0
   Ethernet40   1008185          0              0             0           0
   Ethernet41   1007467          0              0             0           0
   Ethernet42     82925          0              0             0           0
Port-Channel1   1008551          0              0             0           0
Port-Channel2    334854     678589              0             0           3
Port-Channel3   1010420          4              0             0           0

switch>
```

show spanning-tree interface

The **show spanning-tree interface** command displays spanning tree protocol information for the specified interface.

Command Mode

EXEC

Command Syntax

show spanning-tree interface *INT_NAME* [***INFO_LEVEL***]

Parameters

- ***INT_NAME*** Interface type and number. Values include
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **peerethernet***e_num* Ethernet interface specified by *e_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **peerport-channel***p_num* Port-Channel Interface specified by *p_num*.
- ***INFO_LEVEL*** specifies level of detail provided by the output. Options include:
  - <no parameter> command displays a table of STP data for the specified interface.
  - **detail** command displays a data block for the specified interface.

Examples

- This command displays an STP table for Ethernet 5 interface.

```
switch>show spanning-tree interface ethernet 5
Instance         Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
MST0             designated forwarding 20000     128.5    P2p
switch>
```

- This command displays a data block for Ethernet interface 5.

```
switch>show spanning-tree interface ethernet 5 detail
 Port 5 (Ethernet5) of MST0 is designated forwarding
   Port path cost 20000, Port priority 128, Port Identifier 128.5.
   Designated root has priority 32768, address 001c.7301.07b9
   Designated bridge has priority 32768, address 001c.7304.195b
   Designated port id is 128.5, designated path cost 1999 (Ext) 0 (Int)
   Timers: message age 1, forward delay 15, hold 20
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default, Internal
   BPDU: sent 1008766, received 0, taggedErr 0, otherErr 0, rateLimiterCount 0
   Rate-Limiter: enabled, Window: 10 sec, Max-BPDU: 400

switch>
```

show spanning-tree mst

The **show spanning-tree mst** command displays configuration and state information for Multiple Spanning Tree Protocol (MST) instances.

Command Mode

EXEC

Command Syntax

show spanning-tree mst [***INSTANCE***] [***INFO_LEVEL***]

Parameters

- ***INSTANCE*** – MST instance for which command displays information. Options include
  - — <no parameter> all MST instances.
  - — *mst_inst* MST instance number. Value of *mst_inst* ranges from 0 to 4094.
- ***INFO_LEVEL*** – type and amount of information in the output. Options include:
  - — <no parameter> output is interface data in tabular format.
  - — **detail** output is a data block for each interface.

Examples

- This command displays interface data blocks for MST instance 3.

```
switch>show spanning-tree mst 3 detail
##### MST3    vlans mapped:    3
Bridge        address 0011.2233.4402  priority      32771 (32768 sysid 3)
Root          address 0011.2233.4401  priority      32771 (32768 sysid 3)

Ethernet1 of MST3 is root forwarding
Port info             port id          128.1  priority    128  cost       2000
Designated root       address 0011.2233.4401  priority  32768  cost          0
Designated bridge     address 0011.2233.4401  priority  32768  port id   128.1

Ethernet2 of MST3 is alternate discarding
Port info             port id          128.2 priority    128  cost       2000
Designated root       address 0011.2233.4401 priority  32768  cost          0
Designated bridge     address 0011.2233.4401 priority  32768  port id   128.2

Ethernet3 of MST3 is designated forwarding
Port info             port id          128.3 priority    128  cost       2000
Designated root       address 0011.2233.4401 priority  32768  cost       2000
Designated bridge     address 0011.2233.4402 priority  32768  port id   128.3
```

- This command displays interface tables for all MST instances.

```
switch>show spanning-tree mst
##### MST0    vlans mapped:    1,4-4094
Bridge        address 0011.2233.4402  priority      32768 (32768 sysid 0)
Root          address 0011.2233.4401  priority      32768 (32768 sysid 0)
Regional Root address 0011.2233.4401  priority      32768 (32768 sysid 0)

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et1              root       forwarding 2000      128.1    P2p
Et2              alternate  discarding 2000      128.2    P2p
Et3              designated forwarding 2000      128.3    P2p
Et4              designated forwarding 2000      128.4    P2p

##### MST2 vlans mapped: 2
Bridge        address 0011.2233.4402  priority       8194 (8192 sysid 2)
Root          this switch for MST2

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et1              designated forwarding 2000      128.1    P2p
Et2              designated forwarding 2000      128.2    P2p
Et3              designated forwarding 2000      128.3    P2p
Et4              designated forwarding 2000      128.4    P2p

##### MST3 vlans mapped: 3
Bridge        address 0011.2233.4402  priority      32771 (32768 sysid 3)
Root          address 0011.2233.4401  priority      32771 (32768 sysid 3)

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et1              root       forwarding 2000      128.1    P2p
Et2              alternate  discarding 2000      128.2    P2p
Et3              designated forwarding 2000      128.3    P2p
Et4              designated forwarding 2000      128.4    P2p
```

show spanning-tree mst configuration

The **show spanning-tree mst configuration** command displays information about the MST region's VLAN-to-instance mapping. The command provides two display options:

- **default** displays a table that lists the instance to VLAN map.
- **digest** displays the configuration digest.

The configuration digest is a 16-byte hex string calculated from the md5 encoding of the VLAN-to-instance mapping table. Switches with identical mappings have identical digests.

Command Mode

EXEC

Command Syntax

show spanning-tree mst configuration [***INFO_LEVEL***]

Parameters

- *INFO_LEVEL* specifies data provided by the output. Options include:
  - <no parameter> command displays VLAN-to-instance map
  - **digest** command displays the MST configuration digest

Examples

- This command displays the MST region's VLAN-to-instance map.

```
switch>show spanning-tree mst configuration
Name      []
Revision  0    Instances configured 3

Instance  Vlans mapped
--------  ------------------------------------------------------------------------
0         1,4-4094
2         2
3         3
------------------------------------------------------------------------------
switch>
```

- This command displays the MST region's configuration digest.

```
switch>show spanning-tree mst configuration digest
Name      []
Revision  0    Instances configured 1
Digest         0xAC36177F50283CD4B83821D8AB26DE62
switch>
```

show spanning-tree mst interface

The **show spanning-tree mst interface** command displays a Multiple Spanning Tree Protocol (MSTP) information for a specified interface on the specified MST instances.

Command Mode

EXEC

Command Syntax

show spanning-tree mst [***INSTANCE***] **interface** ***INT_NAME*** [***INFO_LEVEL***]

Parameters

- ***INSTANCE*** MST instance for which command displays information. Options include
  - <no parameter> all MST instances.
  - *mst_inst* denotes single MST instance. Value of *mst_inst* ranges from 0 to 4094.
- ***INT_NAME*** Interface type and number. Values include
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **peerethernet***e_num* Ethernet interface specified by *e_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **peerport-channel***p_num* Port-Channel Interface specified by *p_num*.
- ***INFO_LEVEL*** specifies level of detail provided by the output. Options include:
  - <no parameter> command displays a table of STP instance data for the specified interface
  - **detail** command displays a data block for all specified instance-interface combinations.

Examples

- This command displays an table of STP instance data for Ethernet 1 interface:

```
switch>show spanning-tree mst interface ethernet 1
Ethernet1 of MST0 is root forwarding
Edge port: no                               bpdu guard: disabled
Link type: point-to-point
Boundary : Internal
Bpdus sent 2120, received 2164, taggedErr 0, otherErr 0

Instance Role Sts Cost      Prio.Nbr Vlans mapped
-------- ---- --- --------- -------- ------------------------------
0        Root FWD 2000      128.1    1,4-4094
2        Desg FWD 2000      128.1    2
3        Root FWD 2000      128.1    3
```

- This command displays blocks of STP instance information for Ethernet 1 interface.

```
switch>show spanning-tree mst 3 interface ethernet 1 detail
Edge port: no                               bpdu guard: disabled
Link type: point-to-point
Boundary : Internal
Bpdus sent 2321, received 2365, taggedErr 0, otherErr 0

Ethernet1 of MST3 is root forwarding
Vlans mapped to MST3 3
Port info             port id          128.1  priority    128 cost        2000
Designated root       address 0011.2233.4401  priority  32768 cost           0
Designated bridge     address 0011.2233.4401  priority  32768 port id    128.1
```

show spanning-tree mst test information

The **show spanning-tree mst test information** displays diagnostic spanning tree protocol information.

Command Mode

EXEC

Command Syntax

show spanning-tree mst test information

Examples

- This command displays diagnostic STP information.

```
switch>show spanning-tree mst test information
bi = MstInfo.BridgeInfo( "dut" )
bi.stpVersion = "rstp"
bi.mstpRegionId = ""
bi.bridgeAddr = "00:1c:73:01:60:17"
si = MstInfo.BridgeStpiInfo( "Mst" )
bi.stpiInfoIs( "Mst", si )
si.cistRoot = Tac.Value( "Stp::BridgeId", priority=32768, systemId=0,
address='00:1c:73:01:60:17' )
si.cistPathCost = 0
bmi = MstInfo.BridgeMstiInfo( "Mst0" )
bmi.bridgeId = Tac.Value( "Stp::BridgeId", priority=32768, systemId=0,
address='00:1c:73:01:60:17' )
bmi.designatedRoot = Tac.Value( "Stp::BridgeId", priority=32768, systemId=0,
address='00:1c:73:01:60:17' )
si.mstiInfoIs( "Mst0", bmi )
bmii = MstInfo.BridgeMstiIntfInfo( "Mst0", "Ethernet15" )
bmii.portId = Tac.Value( "Stp::PortId",
                    portPriority=128, portNumber=15 )
bmii.role = "designated"
bmii.operIntPathCost = 2000
bmii.fdbFlush = 1
bmi.mstiIntfInfoIs( "Ethernet15", bmii )
bii = MstInfo.BridgeIntfInfo( "Ethernet15" )
bii.operExtPathCost = 2000
si.intfInfoIs( "Ethernet15", bii )
bmii = MstInfo.BridgeMstiIntfInfo( "Mst0", "Port-Channel10" )
bmii.portId = Tac.Value( "Stp::PortId",
                    portPriority=128, portNumber=101 )
bmii.role = "designated"
bmii.operIntPathCost = 1999
bmii.fdbFlush = 1
bmi.mstiIntfInfoIs( "Port-Channel10", bmii )
bii = MstInfo.BridgeIntfInfo( "Port-Channel10" )
bii.operExtPathCost = 1999
si.intfInfoIs( "Port-Channel10", bii )
switch>
```

show spanning-tree root

The **show spanning-tree root** command displays the Bridge-ID, cost to the root bridge, root port, and the root bridge timer settings for all instances.

Command Mode

EXEC

Command Syntax

show spanning-tree root [***INFO_LEVEL***]

Parameters

- *INFO_LEVEL* specifies output format. Options include:
  - <no parameter> output displays data in tabular format.
  - **detail** output displays a data block for each instance.

Examples

- This command displays a table of root bridge information.

```
switch>show spanning-tree root
                  Root ID          Root    Hello  Max  Fwd
Instance     Priority    MAC addr   Cost    Time   Age  Dly  Root Port
----------   ------------------- --------- -----  ---  ---  ------------
MST0         32768 001c.7301.23de         0    2    20  15   Po937
MST101       32869 001c.7301.23de      3998    0     0   0   Po909
MST102       32870 001c.7301.23de      3998    0     0   0   Po911
switch>
```

- This command displays root bridge data blocks for each MSTP instance.

```
switch>show spanning-tree root detail
MST0
MST0
  Root ID    Priority    32768
             Address     001c.7301.23de
             Cost        0 (Ext) 3998 (Int)
             Port        100 (Port-Channel937)
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec
MST101
  Root ID    Priority    32869
             Address     001c.7301.23de
             Cost        3998
             Port        107 (Port-Channel909)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec
MST102
  Root ID    Priority    32870
             Address     001c.7301.23de
             Cost        3998
             Port        104 (Port-Channel911)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec
switch>
```

show spanning-tree topology status

The **show spanning-tree topology status** command displays the forwarding state of ports on the specified VLANs.

Command Mode

EXEC

Command Syntax

show spanning-tree topology [***VLAN_NAME***] **status** [***INFO_LEVEL***]

Parameters

- ***VLAN_NAME*** specifies the VLANs that the output displays. Options include:
  - — <no parameter> output includes all VLANs.
  - — **vlan** output includes all VLANs.
  - — **vlan** *v_num* command includes specified VLAN; *v_num* ranges from 1 to 4094.
- ***INFO_LEVEL*** specifies information provided by output. Options include:
  - — <no parameter> output lists forwarding state of interfaces.
  - — **detail** output lists forwarding state and change history of interfaces.

Examples

- This command displays forwarding state for ports mapped to all VLANs.

```
switch>show spanning-tree topology status
Topology: Cist
  Mapped Vlans: 1-4,666,1000-1001,1004-1005
  Cpu:               forwarding
  Ethernet2:         forwarding
  Ethernet3:         forwarding
  Ethernet4:         forwarding
  Ethernet5:         forwarding
  Ethernet6:         forwarding
  Ethernet8:         forwarding
  Ethernet10:        forwarding
  Port-Channel1:     forwarding
  Port-Channel2:     forwarding
  Port-Channel3:     forwarding

switch>
```

- This command displays forwarding state and history for ports mapped to VLAN 1000.

```
switch>show spanning-tree topology vlan 1000 status detail
Topology: Cist
  Mapped Vlans: 1000
  Cpu:               forwarding (1 changes, last 23 days, 22:54:43 ago)
  Ethernet2:         forwarding (3 changes, last 23 days, 22:48:59 ago)
  Ethernet4:         forwarding (3 changes, last 10 days, 19:54:17 ago)
  Ethernet5:         forwarding (3 changes, last 23 days, 22:54:38 ago)
  Ethernet6:         forwarding (3 changes, last 19 days, 15:49:10 ago)
  Ethernet10:        forwarding (3 changes, last 9 days, 7:37:05 ago)
  Port-Channel1:     forwarding (3 changes, last 23 days, 22:54:34 ago)
  Port-Channel3:     forwarding (5 changes, last 21 days, 4:56:41 ago)

switch>
```

spanning-tree bpdufilter

The **spanning-tree bpdufilter** command controls bridge protocol data unit (BPDU) filtering on the configuration mode interface. BPDU filtering is disabled by default.

Ports with BPDU filtering enabled drop inbound BPDUs and do not send BPDUs. Enabling BPDU filtering on a port not connected to a host can result in loops as the port continues forwarding data while ignoring inbound BPDU packets.

- **spanning-tree bpdufilter enabled** enables BPDU filtering.
- **spanning-tree bpdufilter disabled** disables BPDU filtering by removing the **spanning-tree bpdufilter** command from *running-config*.

The BPDU filter default setting for portfast ports is configured by the **spanning-tree portfast bpdufilter default** command; BPDU filter is disabled by default on all non-portfast ports.

The **no spanning-tree bpdufilter** and **default spanning-tree bpdufilter** commands restore the global BPDU filter setting on the configuration mode interface by removing the corresponding **spanning-tree bpdufilter** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree bpdufilter FILTER_STATUS
no spanning-tree bpdufilter
default spanning-tree bpdufilter
```

Parameters

- *FILTER_STATUS* BPDU filtering status. Options include:
  - **enabled** BPDU filter is enabled on the interface.
  - **disabled** BPDU filter is disabled on the interface.

Examples

- This command enables BPDU filtering on Ethernet 5 interface.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree bpdufilter enabled
switch(config-if-Et5)#
```

spanning-tree bpduguard

The **spanning-tree bpduguard** command controls BPDU guard on the configuration mode interface. A BPDU guard-enabled port is disabled when it receives a BPDU packet. Disabled ports differ from blocked ports in that they are re-enabled only through manual intervention.

The BPDU guard default setting for portfast ports is configured by the **spanning-tree portfast bpduguard default** command; BPDU guard is disabled by default on all non-portfast ports.

- **spanning-tree bpduguard enable** enables BPDU guard on the interface.
- **spanning-tree bpduguard disable** disables BPDU guard on the interface.

The **no spanning-tree bpduguard** and **default spanning-tree bpduguard** commands restore the global BPDU guard setting on the configuration mode interface by removing the corresponding **spanning-tree bpduguard** command from ***running-config.***

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree bpduguard GUARD_ACTION
no spanning-tree bpduguard
default spanning-tree bpduguard
```

Parameters

- *GUARD_ACTION* BPDU guard setting. Options include:
  - **enabled** BPDU guard is enabled on the interface.
  - **disabled** BPDU guard is disabled on the interface.

Examples

- These commands enable BPDU guard on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree bpduguard enabled
switch(config-if-Et5)
```

spanning-tree bpduguard rate-limit count (global)

The **spanning-tree bpduguard rate-limit count** command sets the maximum BPDU reception rate (quantity per interval) for ports not covered by a **spanning-tree bpduguard rate-limit count (interface)** command.

- The default quantity is 10 times the number of VLANs.
- The default interval is the hello time (**spanning-tree hello-time**).

BPDU rate limiting restricts the number of BPDUs that ports with BPDU guard or BPDU filter disabled can process during a specified interval. Ports discard BPDUs it receives in excess of the specified limit. BPDU rate limiting is enabled or disabled by **spanning-tree bpduguard rate-limit enable / disable** commands.

The **no spanning-tree bpduguard rate-limit count** and **default spanning-tree bpduguard rate-limit count** commands restore the global setting to its default value by removing the **spanning-tree bpduguard rate-limit count** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree bpduguard rate-limit count max_bpdu [TIMER]
no spanning-tree bpduguard rate-limit count
default spanning-tree bpduguard rate-limit count
```

Parameters

- *max_bpdu* BPDU quantity. Value ranges from 1 to 20000.
- ***TIMER*** BPDU reception interval (seconds). Options include
  - <no parameter> reception interval defaults to ***hello-time***.
  - **interval** *period* Value of *period* ranges from 1 to 15.

Example

- This command configures the global rate limit as 5000 BPDUs per four second period.

```
switch(config)#spanning-tree bpduguard rate-limit count 5000 interval 4
switch(config)#
```

spanning-tree bpduguard rate-limit count (interface)

The **spanning-tree bpduguard rate-limit count** command configures the maximum BPDU reception rate for the configuration mode interface. The default rate limit is specified by the **spanning-tree bpduguard rate-limit count (global)** command.

BPDU rate limiting restricts the number of BPDUs that ports with BPDU guard or BPDU filter disabled can process during a specified interval. Ports discard BPDUs it receives in excess of the specified limit. BPDU rate limiting is enabled or disabled by **spanning-tree bpduguard rate-limit enable / disable** commands.

The **no spanning-tree bpduguard rate-limit count** and **default spanning-tree bpduguard rate-limit count** commands restore the interface value to the global setting by removing the corresponding **spanning-tree bpduguard rate-limit count** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
spanning-tree bpduguard rate-limit count max_bpdu [TIMER]
no spanning-tree bpduguard rate-limit count
default spanning-tree bpduguard rate-limit count
```

Parameters

- *max_bpdu* BPDU quantity. Value ranges from 1 to 20,000.
- ***TIMER*** BPDU reception interval (seconds). Options include
  - <no parameter> reception interval defaults to ***hello-time***.
  - **interval** *period* Value of *period* ranges from 1 to 15.

Example

- These commands configure rate limit as 7500 BPDUs per 8 second period on Ethernet 2.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#spanning-tree bpduguard rate-limit count 7500 interval 8
switch(config-if-Et2)#
```

spanning-tree bpduguard rate-limit default

The **spanning-tree bpduguard rate-limit default** command configures the global BPDU rate limit setting. The global BPDU rate limit setting provides the default for individual ports whose configuration does not include a **spanning-tree bpduguard rate-limit enable / disable** command. The default global setting is ***enabled***.

BPDU rate limiting restricts the number of BPDUs that ports with BPDU guard or BPDU filter disabled can process during a specified interval. Ports discard BPDUs it receives in excess of the specified limit. BPDU rate limits are established by **spanning-tree bpduguard rate-limit count (global)** commands.

The **no spanning-tree bpduguard rate-limit default** sets the global BPDU rate limit setting to ***disabled***. The **spanning-tree bpduguard rate-limit default** and **default spanning-tree bpduguard rate-limit default** commands restore the default global rate limit ssetting to ***enabled*** by removing the **no spanning-tree bpduguard rate-limit default** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree bpduguard rate-limit default
no spanning-tree bpduguard rate-limit default
default spanning-tree bpduguard rate-limit default
```

Example

- This command enables rate limiting on all ports not covered by an interface rate limit command.

```
switch(config)#spanning-tree bpduguard rate-limit default
switch(config)#
```

spanning-tree bpduguard rate-limit enable / disable

These commands enable and disable BPDU rate limiting on the configuration mode interface:

- **spanning-tree bpduguard rate-limit enable** enables BPDU rate limiting.
- **spanning-tree bpduguard rate-limit disable** disables BPDU rate limiting.

The **spanning-tree bpduguard rate-limit default** command enables BPDU rate limiting on all ports not configured with a **spanning-tree bpduguard rate-limit** command.

BPDU rate limiting restricts the number of BPDUs that ports with BPDU guard or BPDU filter disabled can process during a specified interval. Ports discard BPDUs it receives in excess of the specified limit. BPDU rate limits are established by **spanning-tree bpduguard rate-limit count (interface)** commands.

The **no spanning-tree bpduguard rate-limit** and **default spanning-tree bpduguard rate-limit** commands restore the global rate limit setting on the configuration mode interface by removing the corresponding **spanning-tree bpduguard rate-limit** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
spanning-tree bpduguard rate-limit enable
spanning-tree bpduguard rate-limit disable
no spanning-tree bpduguard rate-limit
default spanning-tree bpduguard rate-limit
```

Example

- These commands enable rate limiting on Ethernet 15.

```
switch(config)#interface ethernet 15
switch(config-if-Et15)#spanning-tree bpduguard rate-limit enable
switch(config-if-Et15)#
```

spanning-tree bridge assurance

The **spanning-tree bridge assurance** command enables bridge assurance on all ports with a port type of ***network***. Bridge assurance protects against unidirectional link failure, other software failure, and devices that quit running a spanning tree algorithm.

Bridge assurance is available only on spanning tree ***network*** ports on point-to-point links. Both ends of the link must have bridge assurance enabled. If the device on one side of the link has bridge assurance enabled and the device on the other side either does not support bridge assurance or does not have it enabled, the bridge assurance enabled port is blocked.

The **no spanning-tree bridge assurance** command disables bridge assurance.

The **spanning-tree bridge assurance** and **default spanning-tree bridge assurance** commands restore the default behavior by removing the **no spanning-tree bridge assurance** command from ***running-config***. Only the **no** form of this command is visible in ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree bridge assurance
no spanning-tree bridge assurance
default spanning-tree bridge assurance
```

Examples

- This command enables bridge assurance on the switch.

```
switch(config)#spanning-tree bridge assurance
switch(config)#
```

spanning-tree cost

The **spanning-tree cost** command configures the path cost of the configuration mode interface. Cost values range from 1 to 200000000 (200 million). The default cost depends on the interface speed:

- 1 gigabit interface: cost = 20000
- 10 gigabit interface: cost = 2000

The **spanning-tree cost** command provides a mode option:

- RST instance cost is configured by not including a mode.
- MST instance 0 cost is configured by not including a mode or with the **mst** mode option.
- MST instance cost is configured with the **mst** mode option.
- Rapid-PVST VLAN cost is configured with the **vlan** mode option.

The **no spanning-tree cost** and **default spanning-tree cost** commands restore the default cost on the configuration mode interface by removing the corresponding **spanning-tree cost** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree MODE cost value
no spanning-tree MODE cost
default spanning-tree MODE cost
```

Parameters

- *MODE* specifies the spanning tree instances for which the cost is configured. Values include:
  - — <no parameter> RST instance or MST instance 0.
  - — **mst** *m_range* specified MST instances. *m_range* formats include a number, number range, or comma-delimited list of numbers and ranges. Instance numbers range from 0 to 4094.
  - — **vlan** *v_range* specified Rapid-PVST instances. *v_range* formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from 1 to 4094.
- *value* path cost assigned to interface. Values range from 1 to 200000000 (200 million). Default values are 20000 (1 G interfaces) or 2000 (10 G interfaces).

Examples

- These commands configure a port cost of 25000 for Ethernet interface 5 when configured as an RST port or a port in MST instance 0.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning tree cost 25000
```

- This command configures a port cost of 30000 for Ethernet interface 5 when configured as a port in MST instance 200.

```
switch(config-if-Et5)#spanning tree mst 200 cost 30000
```

- This command configures a port cost of 100000 for Ethernet interface 5 when configured as a port in VLANs 200-220.

```
switch(config-if-Et5)#spanning tree vlan 200-220 cost 100000
switch(config-if-Et5)#
```

spanning-tree forward-time

The **spanning-tree forward-time** command configures the forward delay timer. Forward delay is the time that a port is in listening and learning states before it begins forwarding data packets.

The switch inserts the forward delay timer value in BPDU packets it sends as the root bridge. The forward delay value ranges from 4 to 30 seconds with a default of 15 seconds.

The **no spanning-tree forward-time** and **default spanning-tree forward-time** commands restore the forward delay timer default of 15 seconds by removing the **spanning-tree forward-time** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree forward-time period
no spanning-tree forward-time
default spanning-tree forward-time
```

Parameters

- *period* forward delay timer (seconds). Value ranges from 4 to 30. Default is 15.

Examples

- This command sets the forward delay timer value to 25 seconds.

```
switch(config)#spanning-tree forward-time 25
switch(config)#
```

spanning-tree guard

The **spanning-tree guard** command enables root guard or loop guard on the configuration mode interface. The **spanning-tree loopguard default** command configures the global loop guard setting.

- Root guard prevents a port from becoming a root or blocked port. A root guard port that receives a superior BPDU transitions to the root-inconsistent (blocked) state.
- Loop guard protects against loops resulting from unidirectional link failures on point-to-point links by preventing non-designated ports from becoming designated ports. When loop guard is enabled, a root or blocked port transitions to loop-inconsistent (blocked) state if it stops receiving BPDUs from its designated port. The port returns to its prior state when it receives a BPDU.

The **no spanning-tree guard** and **default spanning-tree guard** commands sets the configuration mode interface to the global loop guard mode by removing the **spanning-tree guard** statement from *running-config*. The **spanning-tree guard none** command disables loop guard and root guard on the interface, overriding the global setting.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree guard PORT_MODE
no spanning-tree guard
default spanning-tree guard
```

Parameters

- *PORT_MODE* the port mode. Options include:
  - **loop** enables loop guard on the interface.
  - **root** enables root guard on the interface.
  - **none** disables root guard and loop guard.

Examples

- This command enables root guard on Ethernet 5 interface.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree guard root
switch(config-if-Et5)#
```

spanning-tree hello-time

The **spanning-tree hello-time** command configures the hello time, which specifies the transmission interval between consecutive bridge protocol data units (BPDU) that the switch sends as a root bridge. The hello time is also inserted in outbound BPDUs.

This hello time ranges from 0.2 seconds to 10 seconds with a default of 2 seconds.

The **no spanning-tree hello-time** and **default spanning-tree hello-time** commands restore the hello time default of 2 seconds by removing the **spanning-tree hello-time** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree hello-time period
no spanning-tree hello-time
default spanning-tree hello-time
```

Parameters

- *period* hello-time (milliseconds). Value ranges from 200 to 10000. Default is 2000.

Examples

- This command configures a hello-time of one second.

```
switch(config)#spanning-tree hello-time 1000
switch(config)#
```

spanning-tree link-type

The **spanning-tree link-type** command specifies the configuration mode interface's link type, which is normally derived from the port's duplex setting. The default setting depends on a port's duplex mode:

- full-duplex ports are ***point-to-point***.
- half-duplex ports are ***shared***.

RSTP can only achieve rapid transition to the forwarding state on edge ports and point-to-point links.

The **no spanning-tree link-type** and **default spanning-tree link-type** commands restore the default link type on the configuration mode interface by removing the corresponding **spanning-tree link-type** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree link-type TYPE
no spanning-tree link-type
default spanning-tree link-type
```

Parameters

- ***TYPE*** link type of the configuration mode interface. Options include:
  - **point-to-point**
  - **shared**

Examples

- This command configures Ethernet 5 interface as a shared port.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree link-type shared
switch(config-if-Et5)#
```

spanning-tree loopguard default

The **spanning-tree loopguard default** command configures the global loop guard setting as ***enabled***. Ports not covered by a **spanning-tree guard** command use the global loop guard setting. Loop guard prevents blocked or root ports from becoming a designated port due to failures resulting in a unidirectional link. The **spanning-tree guard** interface configuration statement overrides the global setting for a specified interface. The default global loop guard setting is ***disabled***.

The **no spanning-tree loopguard default** and **default spanning-tree loopguard default** commands restore the global loop guard setting of ***disabled*** by removing the **spanning-tree loopguard default** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree loopguard default
no spanning-tree loopguard default
default spanning-tree loopguard default
```

Examples

- This command enables loop guard as the default on all switch ports.

```
switch(config)#spanning-tree loopguard default
switch(config)#
```

spanning-tree max-age

The **spanning-tree max-age** command configures the switch's max age timer, which specifies the max age value that the switch inserts in outbound BPDU packets it sends as a root bridge. The max-age time value ranges from 6 to 40 seconds with a default of 20 seconds.

Max age is the interval, specified in the BPDU, that BPDU data remains valid after its reception. The bridge recomputes the spanning tree topology if it does not receive a new BPDU before max age expiry.

The **no spanning-tree max-age** and **default spanning-tree max-age** commands restore the max-age default of 20 seconds by removing the **spanning-tree max-age** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree max-age period
no spanning-tree max-age
default spanning-tree max-age
```

Parameters

- *period* max age period (seconds). Value ranges from 6 to 40. Default is 20.

Examples

- This command sets the max age timer value to 25 seconds.

```
switch(config)#spanning-tree max-age 25
switch(config)#
```

spanning-tree max-hops

The **spanning-tree max-hops** command specifies the max hop setting that the switch inserts into BPDUs that it sends out as the root bridge. The max hop setting determines the number of bridges in an MST region that a BPDU can traverse before it is discarded. The max-hop value ranges from 1 to 255 with a default of 20.

The **no spanning-tree max-hops** and **default spanning-tree max-hops** commands restore the max-hops setting to its default value of 20 by removing the **spanning-tree max-hops** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree max-hops ports
no spanning-tree max-hops
default spanning-tree max-hops
```

Parameters

- *ports* max hops (bridges). Value ranges from 1 to 255. Default is 20.

Examples

- This command sets the max hop value to 40.

```
switch(config)#spanning-tree max-hop 40
switch(config)#
```

spanning-tree mode

The **spanning-tree mode** command specifies the spanning tree protocol version that the switch runs. The default mode is Multiple Spanning Tree (mstp).

The **no spanning-tree mode** and **default spanning-tree mode** commands restore the default spanning tree protocol version.

Caution The **spanning-tree mode** command may disrupt user traffic. When the switch starts a different STP version, all spanning-tree instances are stopped, then restarted in the new mode.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree mode VERSION
no spanning-tree mode
default spanning-tree mode
```

Parameters

- *VERSION* spanning tree version that the switch runs. Options include:
  - **mstp** multiple spanning tree protocol described in the IEEE 802.1Q-2005 specification and originally specified in the IEEE 802.1s specification.
  - **rstp** rapid spanning tree protocol described in the IEEE 802.1D-2004 specification and originally specified in the IEEE 802.1w specification.
  - **rapid-pvst** rapid per-VLAN spanning tree protocol described in the IEEE 802.1D-2004 specification and originally specified in the IEEE 802.1w specification.
  - **backup** disables STP and enables switchport interface pairs configured with the **switchport backup interface** command.
  - **none** disables STP. The switch does not generate STP packets. Each switchport interface forwards data packets to all connected ports and forwards STP packets as multicast data packets on the VLAN where they are received.

Examples

- This command configures the switch to run multiple spanning tree protocol.

```
switch(config)#spanning-tree mode mstp
switch(config)#
```

spanning-tree mst configuration

The **spanning-tree mst configuration** command places the switch in MST-configuration mode, which is the group change mode where MST region parameters are configured.

Changes made in a group change mode are saved by leaving the mode through the **exit** command or by entering another configuration mode. To discard changes from the current edit session, leave the mode with the **abort** command.

These commands are available in MST-configuration mode:

- **abort (mst-configuration mode)**
- **exit (mst-configuration mode)**
- **instance**
- **name (mst-configuration mode)**
- **revision (mst-configuration mode)**
- **show (mst-configuration mode)**

The **no spanning-tree mst configuration** and **default spanning-tree mst configuration** commands restore the MST default configuration.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree mst configuration
no spanning-tree mst configuration
default spanning-tree mst configuration
```

Examples

- This command enters MST configuration mode.

```
switch(config)#spanning-tree mst configuration
switch(config-mst)#
```

- This command exits MST configuration mode, saving MST region configuration changes to *running-config*.

```
switch(config-mst)#exit
switch(config)#
```

- This command exits MST configuration mode without saving MST region configuration changes to *running-config*.

```
switch(config-mst)#abort
switch(config)#
```

spanning-tree portchannel guard misconfig

The **spanning-tree portchannel guard misconfig** command enables the switch to detect misconfigured port channels that may cause network loops by monitoring inbound BPDUs. When a port channel receives 75 inconsistent BPDUs within 30 seconds, the switch error disables the port. When a port channel receives 5 BPDUs with the same source BPDU during the 30 second measurement interval, the error counter is reset and the port continues normal port channel operation. Misconfigured port channel detection is enabled by default.

The **no spanning-tree portchannel guard misconfig** disables the misconfigured port channel detection.

The **default spanning-tree portchannel guard misconfig** command restores port channel detection by removing the **no spanning-tree portchannel guard misconfig** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree portchannel guard misconfig
no spanning-tree portchannel guard misconfig
default spanning-tree portchannel guard misconfig

spanning-tree etherchannel guard misconfig
no spanning-tree etherchannel guard misconfig
default spanning-tree etherchannel guard misconfig
```

Guidelines

The **spanning-tree portchannel guard misconfig** and **spanning-tree etherchannel guard misconfig** commands are equivalent.

Examples

- This command disables port channel misconfiguration detection on the switch.

```
switch(config)#no spanning-tree portchannel guard misconfig
switch(config)#
```

- This command enaables port channel misconfiguration detection on the switch.

```
switch(config)#spanning-tree portchannel guard misconfig
switch(config)#
```

spanning-tree portfast

The **spanning-tree portfast** command programs configuration mode ports to immediately enter forwarding state when they establish a link, bypassing listening and learning states. PortFast ports are included in spanning tree topology calculations and can enter blocking state. This command overrides the **spanning-tree portfast auto** command.

The **no spanning-tree portfast** and **default spanning-tree portfast** commands remove the corresponding **spanning-tree portfast** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree portfast
no spanning-tree portfast
default spanning-tree portfast
```

Example

- This command unconditionally enables portfast on Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast
switch(config-if-Et5)#
```

spanning-tree portfast auto

The **spanning-tree portfast auto** command enables auto-edge detection on the configuration mode interface. When auto-edge detection is enabled, the port is configured as an edge port if it does not receive a new BPDU before the current BPDU expires. Auto-edge detection is enabled by default. The **spanning-tree portfast** command, when configured, has priority over this command.

The **no spanning-tree portfast auto** command disables auto-edge port detection. This command is removed from *running-config* with the **spanning-tree portfast auto** and **default spanning-tree portfast auto** commands.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree portfast auto
no spanning-tree portfast auto
default spanning-tree portfast auto
```

Example

- This command enables auto-edge detection on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast auto
switch(config-if-Et5)#
```

spanning-tree portfast bpdufilter default

The **spanning-tree portfast bpdufilter default** command configures the global BPDU filter setting as ***enabled***. Ports not covered by a **spanning-tree bpdufilter** command use the global BPDU filter setting.

A BPDU filter enabled port drops inbound BPDUs and do not send BPDUs. Enabling BPDU filtering on a port not connected to a host can result in loops as the port continues forwarding data while ignoring inbound BPDU packets.

The global BPDU filter default setting ***disabled***. The **spanning-tree bpdufilter** interface command takes precedence over the global setting for individual ports.

The **no spanning-tree portfast bpdufilter default** and **default spanning-tree portfast bpdufilter default** commands restore the global BPDU filter setting of ***disabled*** by removing the **spanning-tree portfast bpdufilter default** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree portfast bpdufilter default
no spanning-tree portfast bpdufilter default
default spanning-tree portfast bpdufilter default
```

Examples

- This command configures the BPDU filter global setting to ***enabled***.

```
switch(config)#spanning-tree portfast bpdugfilter default
switch(config)#
```

spanning-tree portfast bpduguard default

The **spanning-tree portfast bpduguard default** command configures the global BPDU guard setting as *enabled*. Ports not covered by a **spanning-tree bpduguard** command use the global BPDU guard setting.

BPDU guard disables ports that receive a bridge protocol data unit (BPDU). Disabled ports differ from blocked ports in that they are re-enabled only through manual intervention.

The global BPDU guard setting affects all ports that meet both of the following:

- PortFast is enabled.
- The port is not covered by a **spanning-tree bpduguard** interface command.

The global BPDU guard default setting *disabled*. The **spanning-tree bpduguard** interface command takes precedence over the global setting for individual ports.

The **no spanning-tree portfast bpduguard default** and **default spanning-tree portfast bpduguard default** commands restore the global BPDU guard setting of *disabled* by removing the **spanning-tree portfast bpduguard default** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree portfast bpduguard default
no spanning-tree portfast bpduguard default
default spanning-tree portfast bpduguard default
```

Example

- This command configures the global BPDU guard setting to *enabled*.

```
switch(config)#spanning-tree portfast bpduguard default
switch(config)#
```

spanning-tree portfast <port type>

The **spanning-tree portfast <port-type>** command specifies the STP port mode for the configuration mode interface. Default port mode is ***normal***.

Port modes include:

- ***Edge***: Edge ports connect to hosts and transition to the forwarding state when the link is established, bypassing listening and learning states. An edge port that receives a BPDU becomes a normal port.
- ***Network***: Network ports connect only to switches or bridges and support bridge assurance. Network ports that connect to hosts or other edge devices transition to the blocking state.
- ***Normal***: Normal ports function as normal STP ports and can connect to any type of device.

The **no spanning-tree portfast <port-type>** and **default spanning-tree portfast <port-type>** commands restore the default port mode of normal by removing the corresponding **spanning-tree portfast <port-type>** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree portfast PORT_MODE
no spanning-tree portfast PORT_MODE
default spanning-tree portfast PORT_MODE
```

Parameters

- *PORT_MODE* STP port mode. Options include:
  - **edge**
  - **network**
  - **normal**

  The **normal** option is not available for the **no** and **default** commands.

Related Commands

The **spanning-tree portfast <port-type> command** also affects the **spanning-tree portfast auto** and **spanning-tree portfast** configuration for the configuration mode interface:

- **spanning-tree portfast normal**: **spanning-tree portfast auto** is enabled.
- **spanning-tree portfast edge**: **spanning-tree portfast** is enabled.
- **spanning-tree portfast network**: **spanning-tree portfast auto** is disabled.

Example

- This command configures Ethernet 5 interface as a network port.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree portfast network
switch(config-if-Et5)#
```

spanning-tree port-priority

The **spanning-tree port-priority** command specifies the configuration mode interface's port-priority number. The switch uses this number to determine which interface it places into forwarding mode when resolving a loop. Valid settings are all multiples of 16 between 0 and 240. Default value is 128. Ports with lower numerical priority values are selected over other ports.

The **no spanning-tree port-priority** and default spanning-tree port-priority commands restore the default of 128 for the configuration mode interface by removing the **spanning-tree port-priority** command from ***running-config***.

The **spanning-tree port-priority** command provides a mode option:

- RST instance port-priority is configured by not including a mode.
- MST instance 0 port-priority is configured by not including a mode or with the **mst** mode option.
- MST instance port-priority is configured with the **mst** mode option.
- Rapid-PVST VLAN port-priority is configured with the **vlan** mode option.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
spanning-tree [MODE] port-priority value
no spanning-tree [MODE] port-priority
default spanning-tree [MODE] port-priority
```

Parameters

- ***MODE*** specifies the spanning tree instances for which the cost is configured. Values include:
  - — <no parameter> RST instance or MST instance 0.
  - — **mst** *m_range* specified MST instances. *m_range* formats include a number, number range, or comma-delimited list of numbers and ranges. Instance numbers range from 0 to 4094.
  - — **vlan** *v_range* specified Rapid-PVST instances. *v_range* formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from 1 to 4094.
- *value* bridge priority number. Values range from 0 to 240 and must be a multiple of 16.

Example

- This command sets the port priority of Ethernet 5 interface to 144.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#spanning-tree port-priority 144
switch(config-if-Et5)#
```

spanning-tree priority

The **spanning-tree priority** command configures the bridge priority number. The bridge priority is the four most significant digits of the bridge ID, which is used by spanning tree algorithms to select the root bridge and choose among redundant links. Bridge ID numbers range from 0 to 65535 (16 bits); bridges with smaller bridge IDs are elected over other bridges.

Because bridge priority sets the four most significant bits of the bridge ID, valid settings include all multiples of 4096 between 0 and 61440. Default value is 32768.

The **spanning-tree priority** command provides a mode option:

- RST instance priority is configured by not including a mode.
- MST instance 0 priority is configured by not including a mode or with the **mst** mode option.
- MST instance priority is configured with the **mst** mode option.
- Rapid-PVST VLAN priority is configured with the **vlan** mode option.

The **no spanning-tree priority** and **default spanning-tree priority** commands restore the bridge priority default of 32768 for the specified mode by removing the corresponding **spanning-tree priority** command from ***running-config***.

Another method of adding **spanning-tree priority** commands to the configuration is through the **spanning-tree root** command. Similarly, the **no spanning-tree root** command removes the corresponding **spanning-tree priority** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree [MODE] priority level
no spanning-tree [MODE] priority
default spanning-tree [MODE] priority
```

Parameters

- *MODE* spanning tree instances for which the command configures priority. Options include:
  - — <no parameter> RST instance or MST instance 0.
  - — **mst** *m_range* specified MST instances. *m_range* formats include a number, number range, or comma-delimited list of numbers and ranges. Instance numbers range from 0 to 4094.
  - — **vlan** *v_range* specified Rapid-PVST instances. *v_range* formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from 1 to 4094.
- *level* priority number. Values include multiples of 4096 between 0 and 61440. Default is 32768.

Examples

- This command configures a bridge priority value of 20480 for Rapid-PVST VLANs 20, 24, 28, and 32.

```
switch(config)#spanning-tree vlan 20,24,28,32 priority 20480
switch(config)#
```

- This command configures a bridge priority value of 36864 for the RST instance. When MST is enabled, this command configures a priority of 36864 for MST instance 0.

```
switch(config)#spanning-tree priority 36864
switch(config)#
```

spanning-tree root

The **spanning-tree root** command configures the bridge priority number by adding a **spanning-tree priority** command to the configuration. Parameter settings set the following priority values:

- ***primary*** sets the bridge priority to 8192.
- ***secondary*** sets the bridge priority to 16384.

The bridge priority is the four most significant digits of the bridge ID, which is used by spanning tree algorithms to select the root bridge and choose among redundant links. Bridge ID numbers range from 0 to 65535 (16 bits); bridges with smaller bridge IDs are elected over other bridges.

When no other switch in the network is similarly configured, assigning the primary value to the switch facilitates its selection as the root switch. Assigning the secondary value to the switch facilitates its selection as the backup root in a network that contains one switch with a smaller priority number.

The **spanning-tree root** command provides a mode option:

- RST instance priority is configured by not including a mode.
- MST instance 0 priority is configured by not including a mode or with the **mst** mode option.
- MST instance priority is configured with the **mst** mode option.
- Rapid-PVST VLAN priority is configured with the **vlan** mode option.

The **no spanning-tree root** and **default spanning-tree root** commands restore the bridge priority default of 32768 by removing the corresponding **spanning-tree priority** command from ***running-config***. The **no spanning-tree root**, **no spanning-tree priority, default spanning-tree root** and **default spanning-tree priority** commands perform the same function.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree [MODE] root TYPE
no spanning-tree [MODE] root
default spanning-tree [MODE] root
```

Parameters

- *MODE* specifies the spanning tree instances for which priority is configured. Values include:
  - — <no parameter> RST instance or MST instance 0.
  - — **mst** *m_range* specified MST instances. *m_range* formats include a number, number range, or comma-delimited list of numbers and ranges. Instance numbers range from 0 to 4094.
  - — **vlan** *v_range* specified Rapid-PVST instances. *v_range* formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from 1 to 4094.
- *TYPE* sets the bridge priority number. Values include:
  - — **primary** sets the bridge priority to 8192.
  - — **secondary** sets the bridge priority to 16384.

Examples

- This command configures a bridge priority value of 8192 for Rapid-PVST VLANs 20-36.

```
switch(config)#spanning-tree vlan 20-36 root primary
```

- This command configures a bridge priority value of 16384 for the RSTP instance and MST instance 0.

```
switch(config)#spanning-tree root secondary
```

spanning-tree transmit hold-count

The **spanning-tree transmit hold-count** command specifies the maximum number of BPDUs per second that the switch can send from an interface. Valid settings range from 1 to 10 BPDUs with a default of 6 BPDUs.

The **no spanning-tree transmit hold-count** and **default spanning-tree transmit hold-count** commands restore the transmit hold count default of 6 BPDUs by removing the **spanning-tree transmit hold-count** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
spanning-tree transmit hold-count max_bpdu
no spanning-tree transmit hold-count
default spanning-tree transmit hold-count
```

Parameters

- *max_bpdu* BPDU packets. Value ranges from 1 to 10. Default is 6.

Examples

- This command configures a transmit hold-count of 8 BPDUs.

```
switch(config)#spanning-tree transmit hold-count 8
switch(config)#
```

spanning-tree vlan

The **spanning-tree vlan** command enables spanning-tree on specified VLANs by removing the corresponding **no spanning-tree vlan** statements from ***running-config***. Spanning-tree is enabled on all VLANs by default.

The **no spanning-tree vlan** and default spanning-tree commands disable spanning-tree on the specified interfaces.

Warning Disabling spanning tree is not recommended, even in topologies free of physical loops. Spanning tree guards against configuration mistakes and cabling errors. When disabling VLAN, ensure that there are no physical loops in the VLAN.

Important When disabling spanning tree on a VLAN, ensure that all switches and bridges in the network disable spanning tree for the same VLAN. Disabling spanning tree on a subset of switches and bridges in a VLAN may have unexpected results because switches and bridges running spanning tree will have incomplete information regarding the network's physical topology.

The following spanning-tree global configuration commands provide a vlan option for configuring Rapid-PVST VLAN instances:

- **spanning-tree priority**
- **spanning-tree root**

Command Mode

Global Configuration

Command Syntax

```
spanning-tree vlan v_range
no spanning-tree vlan v_range
default spanning-tree vlan v_range
```

Parameters

- *v_range* VLAN list. Formats include a number, number range, or comma-delimited list of numbers and ranges. VLAN numbers range from 1 to 4094.

Examples

- This command disables spanning-tree on VLANs 200-205

```
switch(config)#no spanning-tree vlan 200-205
switch(config)#
```

- This command enables spanning-tree on VLAN 203

```
switch(config)#spanning-tree vlan 203
switch(config)#
```

switchport backup interface

The **switchport backup interface** command establishes a primary-backup configuration for forwarding VLAN traffic between the command mode interface and a specified interface. The **show interfaces switchport backup** command displays the state of backup interface pairs on the switch.

- the primary interface is the command mode interface.
- the backup interface is the interface specified in the command.

The following guidelines apply to primary and backup interfaces.

- Ethernet and Port Channels can be primary interfaces.
- Ethernet, Port Channel, Management, Loopback, and VLANs can be backup interfaces.
- The primary and backup interfaces can be different interface types.
- Interface pairs should be similarly configured to ensure consistent behavior.
- An interface can be associated with a maximum of one backup interface.
- An interface can back up a maximum of one interface.
- Any Ethernet interface configured in an interface pair cannot be a port channel member.
- The STP mode is backup.
- Static MAC addresses should be configured after primary-backup pairs are established.

When load balancing is not enabled, the primary and backup interfaces cannot simultaneously forward VLAN traffic. When the primary interface is forwarding VLAN traffic, the backup interface drops all traffic. If the primary interface fails, the backup interface forwards VLAN traffic until the primary interface is functional.

The ***prefer vlan*** option balances the load across the primary and backup interfaces. When the command includes the ***prefer vlan*** option, each interface is the primary for a subset of the vlans carried by the pair. When both interfaces are up, prefer option vlans are forwarded on the backup interface and all other configured vlans are carried by the primary interface.

The **no switchport backup interface** and **default switchport backup interface** commands remove the primary-backup configuration for the configuration mode interface.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration

Command Syntax

```
switchport backup interface INT_NAME [BALANCE]
no switchport backup interface
default switchport backup interface
```

Parameters

- ***INT_NAME*** the backup interface. Options include:
  - **ethernet** *e_num* Ethernet interface. *e_num* range depends on switch model.
  - **loopback** *l_num* Loopback interface. *l_num* ranges from 1 to 1000.
  - **management** *m_num* Management interface. *m_num* range depends on switch model.
  - **port-channel** *p_num* Channel group interface. *p_num* ranges from 1 to 1000.
  - **vlan** *v_num* VLAN interface. *v_num* ranges from 1 to 4094.
- ***BALANCE*** VLANs whose traffic is normally handled on the backup interfaces. Values include:
  - <no parameter> backup interface handles no traffic if the primary interface is operating.
  - **prefer vlan** *v_range* list of VLANs whose traffic is handled by backup interface.

Examples

- These commands establish Ethernet interface 7 as the backup port for Ethernet interface 1.

```
switch(config)#interface ethernet 1
switch(config-if-Et1)#switchport backup interface ethernet 7
switch(config-if-Et1)#
```

- These commands configure the following:
  - Ethernet interface 1 as a trunk port that handles VLAN 4 through 9 traffic.
  - Ethernet interface 2 as its backup interface.
  - Ethernet 2 as the preferred interface for VLANs 7 through 9.

```
switch(config-if-Et1)#switchport mode trunk
switch(config-if-Et1)#switchport trunk allowed vlan 4-9
switch(config-if-Et1)#switchport backup Ethernet 2 prefer vlan 7-9
switch(config-if-Et1)#
```

Chapter 16

Quality of Service

This chapter describes Arista's Quality of Service (QoS) implementation, including configuration instructions and command descriptions. Topics covered by this chapter include:

- Section 16.1: Quality of Service Conceptual Overview
- Section 16.2: Quality of Service Configuration Procedures
- Section 16.3: Quality of Service Configuration Commands

16.1 Quality of Service Conceptual Overview

16.1.1 QoS Operation

Quality of Service defines a method of differentiating data streams to provide varying levels of service to the different streams. Criteria determining a packet's priority level include packet field contents and the port where data packets are received. QoS settings are translated into traffic classes, which are then used by switches to manage all traffic flows. Traffic flow management varies with each switch platform.

16.1.1.1 QoS Data Fields

Quality of service decisions are based on the contents of the following packet fields:

- CoS (three bits): Class of service (CoS) is a 3-bit field in Ethernet frame headers using VLAN tagging. The field specifies a priority value between zero and seven. Class of service operates at layer 2.
- DSCP (six bits): Differentiated Service Code Point (DSCP) is a 6-bit field in the VLAN tag of IP packet headers. DSCP operates at layer 3.

16.1.1.2 Port Settings

Ethernet and port channel interfaces support three QoS trust modes:

- CoS Trust: Ports use inbound packet CoS field contents to derive the traffic class.
- DSCP Trust: Ports use inbound packets DSCP field contents to derive the traffic class.
- Untrusted: Ports ignore packet contents, using default CoS values to derive the traffic class.

Ports are associated with default CoS, DSCP, and traffic class settings:

- FM4000 and Trident Platforms: Default CoS and DSCP settings are assigned to all port channel and Ethernet interfaces. Each interface is independently configurable.

- Petra Platforms: One default traffic class is assigned to individual PetraA chips, each of which controls eight Ethernet interfaces. The traffic class value is configurable on each chip. The traffic class value is not configurable on individual interfaces.

16.1.1.3 Traffic Classes

Data stream distribution is based on their traffic classes. Data stream management varies by switch platform. Traffic classes are derived from these data stream, inbound port, and switch attributes:

- CoS field contents
- DSCP field contents
- Inbound port trust setting
- CoS default setting (FM4000 and Trident platforms)
- DSCP default setting (FM4000 and Trident platforms)
- Traffic class default setting (Petra platform)

When a port is configured to derive a data stream's traffic class from the CoS or DSCP value associated with the stream, the traffic class is determined from a conversion map.

- A CoS-traffic class map derives a traffic class from a CoS value.
- A DSCP-traffic class map derives a traffic class from a DSCP value.

Each map entry is configurable through CLI commands. Default maps determine the traffic class value when CLI map entry command are not configured. Default maps vary by switch platform.

16.1.1.4 CoS Rewrite

Switches can rewrite the CoS field for outbound tagged packets. CoS rewrite is enabled by default for packets that are received on DSCP trusted ports and untrusted ports. FM4000 and Trident platform switches provide a command that enables or disables CoS rewrite on these ports. On Petra platform switches, CoS rewrite is always enabled on these ports. The new CoS value is configurable and based on the data stream's traffic class.

CoS rewrite is disabled on all CoS trusted ports.

16.1.1.5 DSCP Rewrite

Switches can rewrite the DSCP field for outbound tagged packets. DSCP rewrite is disabled by default on all ports and always disabled on DSCP trusted ports. On Petra platform switches, DSCP rewrite is always disabled.

FM4000 and Trident platform switches provide a command that enables or disables DSCP rewrite for packets received on CoS trusted ports and untrusted ports. The new DSCP value is configurable and based on the data stream's traffic class.

16.1.1.6 Transmit Queues and Port Shaping

Transmit queues are logical partitions of an Ethernet port's egress bandwidth. Data streams are assigned to queues based on their traffic class, then sent as scheduled by port and transmit settings. Support varies by switch platform. A queue's label determines its priority: Tx-queue 0 has lowest priority.

Parameters that determine transmission schedules include:

- Traffic class-transmit queue mapping: One set of traffic class-transmit queue maps is defined for all interfaces on a switch. The map determines the schedule for transmitting data streams based on traffic class.

  The set of available transmit maps vary by switch platform:

  - FM4000 platform: one map for all unicast and multicast traffic.
  - Trident platform: one map for unicast traffic and one map for multicast traffic.
  - Petra platform: one map for unicast traffic. Queue shaping is not available for multicast traffic.
- Port shaping: Port shaping specifies an Ethernet port's maximum egress bandwidth.
- Queue shaping: Queue shaping specifies a transmit queue's maximum egress bandwidth.

  FM4000 platform switches do not support simultaneous port shaping and queue shaping. Enabling port shaping automatically disables queue shaping, regardless of previously configured commands.
- Queue priority: Queue priority specifies the transmission scheduling algorithm from the transmit queues. The switch defines two queue priority types:
  - Strict Priority: Strict priority queues are serviced in the order of their priority rank, subject to each queue's configured maximum bandwidth. Data is not handled for a queue until all queues with higher priority are emptied or their transmission limit is reached. These queues typically carry low latency real time traffic and require highest available priority.
  - Round Robin: Round robin queues are serviced simultaneously subject to assigned bandwidth percentage and configured maximum bandwitdth. All round robin queues have lower priority than strict priority queues. Round robin queues can be starved by strict priority queues.
- Queue bandwidth allocation: Queue bandwidth allocation specifies the time slice (percentage) assigned to a round robin queue, relative to all other round robin queues.

16.1.2 Arista QoS Implementation

QoS behavior details vary with switch platforms. These sections describe switch QoS behavior for Arista platforms.

16.1.2.1 FM4000 Platform

Traffic Class Derivation Source

Table 16-1 displays the derivation source for a data stream's traffic class on FM4000 switches.

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default CoS (port)	Default CoS (port)	Default DSCP (port)
Untagged IP	Default CoS (port)	Default CoS (port)	DSCP (packet)
Tagged Non-IP	Default CoS (port)	CoS (packet)	Default DSCP (port)
Tagged IP	Default CoS (port)	CoS (packet)	DSCP (packet)

Table 16-1 Traffic Class Derivation Source: FM4000 Platform

Default CoS to Traffic Class Map

Table 16-2 displays the default CoS to Traffic Class map on FM4000 switches.

Derived CoS	Traffic Class
untagged	1
0	1
1	0
2	2
3	3
4	4
5	4
6	5
7	6

Table 16-2 CoS to Traffic Class Map: FM4000 Platform

Default DSCP to Traffic Class Map

Table 16-3 displays the default DSCP to Traffic Class map on FM4000 switches.

Derived DSCP	Traffic Class
0-7	1
8-15	0
16-23	2
24-31	3
32-39	4
40-47	4
48-55	5
56-63	5

Table 16-3 DSCP to Traffic Class Map: FM4000 Platform

Default Traffic Class to Cos Rewrite Value Map

Table 16-4 displays the default Traffic Class to CoS rewrite value map on FM4000 switches.

Traffic Class	CoS
0	1
1	0
2	2
3	3
4	4
5	6
6	7

Table 16-4 Traffic Class to CoS Rewrite Value Map: FM4000 Platform

Default Traffic Class to DSCP Rewrite Value Map

Table 16-5 displays the default Traffic Class to DSCP rewrite value map on FM4000 switches.

Traffic Class	DSCP
0	8
1	0
2	16
3	24
4	32
5	48
6	56

Table 16-5 Traffic Class to DSCP Rewrite Value Map: FM4000 Platform

Default Traffic Class to Transmit Queue Map

Table 16-6 displays the default Traffic Class to Transmit Queue map on FM4000 switches.

Traffic Class	Transmit Queue
0	0
1	1
2	2
3	3
4	4
5	5
6	6

Table 16-6 Traffic Class to Transmit Queue Map: FM4000 Platform

16.1.2.2 Trident Platform

Traffic Class Derivation Source

Table 16-7 displays the derivation source for a data stream's traffic class on Trident switches.

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default CoS (port)	Default CoS (port)	Default DSCP (port)
Untagged IP	Default CoS (port)	Default CoS (port)	DSCP (packet)
Tagged Non-IP	Default CoS (port)	CoS (packet)	Default DSCP (port)
Tagged IP	Default CoS (port)	CoS (packet)	DSCP (packet)

Table 16-7 Traffic Class Derivation Source: Trident Platform

Default CoS to Traffic Class Map

Table 16-8 displays the default CoS to Traffic Class map on Trident switches.

Derived CoS	Traffic Class
untagged	1
0	1
1	0
2	2
3	3
4	4
5	5
6	6
7	7

Table 16-8 CoS to Traffic Class Map: Trident Platform

Default DSCP to Traffic Class Map

Table 16-9 displays the default DSCP to Traffic Class map on Trident switches.

Derived DSCP	Traffic Class
0-7	1
8-15	0
16-23	2
24-31	3
32-39	4
40-47	5
48-55	6
56-63	7

Table 16-9 DSCP to Traffic Class Map: Trident Platform

Default Traffic Class to Cos Rewrite Value Map

Table 16-10 displays the default Traffic Class to CoS rewrite value map on Trident switches.

Traffic Class	CoS
0	1
1	0
2	2
3	3
4	4
5	5
6	6
7	7

Table 16-10 Traffic Class to CoS Rewrite Value Map: Trident Platform

Default Traffic Class to DSCP Rewrite Value Map

Table 16-11 displays the default Traffic Class to DSCP rewrite value map on Trident switches.

Traffic Class	CoS
0	8
1	0
2	16
3	24
4	32
5	40
6	48
7	56

Table 16-11 Traffic Class to DSCP Rewrite Value Map: Trident Platform

Default Traffic Class to Transmit Queue Maps

Table 16-12 displays the default Traffic Class to Unicast and Multicast Transmit Queue map on Trident switches.

	Transmit Queue	
Traffic Class	Unicast	Multicast
0	0	0
1	1	0
2	2	1
3	3	1
4	4	2
5	5	2
6	6	3
7	7	3

Table 16-12 Traffic Class to Transmit Queue Map: Trident Platform

16.1.2.3 Petra Platform

Traffic Class Derivation Source

Table 16-13 displays the derivation source for a data stream's traffic class on Petra switches.

	Untrusted	CoS Trusted	DSCP Trusted
Untagged Non-IP	Default TC (chip)	Default TC (chip)	Default TC (chip)
Untagged IP	Default TC (chip)	Default TC (chip)	DSCP (packet)
Tagged Non-IP	Default TC (chip)	CoS (packet)	Default TC (chip)
Tagged IP	Default TC (chip)	CoS (packet)	DSCP (packet)

Table 16-13 Traffic Class Derivation Source: Petra Platform

Default CoS to Traffic Class Map

Table 16-14 displays the default CoS to Traffic Class map on Petra switches.

Derived CoS	Traffic Class
untagged	1
0	1
1	0
2	2
3	3
4	4
5	5
6	6
7	7

Table 16-14 CoS to Traffic Class Map: Petra Platform

Default DSCP to Traffic Class Map

Table 16-15 displays the default DSCP to Traffic Class map on Petra switches.

Derived DSCP	Traffic Class
0-7	1
8-15	0
16-23	2
24-31	3
32-39	4
40-47	5
48-55	6
56-63	7

Table 16-15 DSCP to Traffic Class Map: Petra Platform

Default Traffic Class to CoS Rewrite Value Map

Table 16-16 displays the default Traffic Class to CoS rewrite value map on Petra switches.

Traffic Class	CoS
0	1
1	0
2	2
3	3
4	4
5	5
6	6
7	7

Table 16-16 Traffic Class to CoS Rewrite Value Map: Petra Platform

Default Traffic Class to Transmit Queue Map

Table 16-17 displays the default Traffic Class to Transmit Queue map on Petra switches.

Traffic Class	Transmit Queue
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Table 16-17 Traffic Class to CoS Transmit Queue Map: Petra Platform

16.2 Quality of Service Configuration Procedures

Implementing QoS on an Arista switch consists of configuring port trust settings, default port settings, default traffic classes, conversion maps, and transmit queues.

- Section 16.2.1: CoS and DSCP Settings
- Section 16.2.2: Traffic Classes
- Section 16.2.3: Transmit Queues and Port Shaping

16.2.1 CoS and DSCP Settings

Configuring Port Trust Settings

The **qos trust** command configures QoS port trust mode for the configuration mode interface. Trust enabled ports classify traffic by examining traffic's CoS or DSCP value. Port-trust default setting is ***cos***.

- The **qos trust cos** command specifies ***cos*** as the port's port-trust mode.
- The **qos trust dscp** command specifies ***dscp*** as the port's port-trust mode.
- The **no qos trust** command specifies ***untrusted*** as the port's port-trust mode.

Examples

- These commands configure trust mode of ***dscp*** for Ethernet interface 7.

```
switch(config)#interface Ethernet 7
switch(config-if-Et7)#qos trust dscp
switch(config-if-Et7)#
```

- These commands configure trust mode of ***untrusted*** for Port Channel interface 23.

```
switch(config)#interface port-channel 23
switch(config-if-Po23)#no qos trust
switch(config-if-Po23)#
```

Configuring Default Port Settings

Ports are associated with default CoS and DSCP settings. Available settings vary with switch platform:

- **FM4000 and Trident Platforms**: Default CoS and DSCP settings are assigned to all port channel and Ethernet interface. Each interface is independently configurable.
  - The **qos cos** command specifies the default class of service (CoS) value of the configuration mode interface. CoS values range from 0 to 7. Default value is 0.
  - The **qos dscp** command specifies the default differentiated services code point (DSCP) value of the configuration mode interface. DSCP values range from 0 to 63. Default value is 0.
- **Petra Platforms**: Each PetraA chip is assigned a default traffic class for the eight Ethernet ports that it controls. Traffic class is configurable on each chip, not individual interfaces.
  - The **platform petraA traffic-class** command specifies the default traffic class for ports controlled by the specified PetraA chip. Default traffic class is a Petra switch feature that replaces **qos cos** and **qos dscp** commands. This command is valid only on Petra switches.

Examples

- This command sets the default DSCP of 44 on Ethernet 7 interface.

```
Switch(config-if-Et7)#qos dscp 44
Switch(config-if-Et7)
```

- This command configures the default CoS of four on Ethernet interface 8.

```
Switch(config-if-Et8)#qos cos 4
Switch(config-if-Et8)#
```

16.2.2 Traffic Classes

Configuring Default Traffic Class

Petra switches assign a default traffic class to all Ethernet interfaces controlled by individual PetraA chips. Traffic class values are configurable for each PetraA chip, not individual interfaces.

The **platform petraA traffic-class** command specifies the default traffic class used by all ports controlled by a specified chip. The default traffic class is an alternative configuration that only Petra switches implement, replacing **qos cos** and **qos dscp** commands. This command is valid only on Petra switches.

Examples

- This command configures the default traffic class to five for the ports 25-32 on linecard 5.

```
switch(config)#platform petraA linecard5-Petra-3 traffic-class 5
switch(config)#
```

- This command configures the default traffic class to three for all ports on linecard 10.

```
switch(config)#platform petraA module 10 traffic-class 3
switch(config)#
```

Mapping CoS to Traffic Class

The **qos map cos** command associates a traffic class to a list of Class of Service (CoS) settings. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the port upon which it is received.

Example

- This command assigns the traffic class of 5 to the classes of service 1, 3, 5, and 7.

```
switch(config)#qos map cos 1 3 5 7 to traffic-class 5
switch(config)#
```

Mapping DSCP to Traffic Class

The **qos map dscp** command associates a traffic class to a set of DSCP values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the port upon which it is received.

Example

- This command assigns the traffic class of three to the DSCP values of 12, 13, 25, and 37.

```
switch(config)#qos map dscp 12 13 25 37 to traffic-class 3
switch(config)#
```

Mapping Traffic Class to CoS

The **qos map traffic-class to cos** command associates a CoS to a list of traffic classes. Multiple commands create a complete map, which the switch uses to fill the CoS field in outbound packets. This map is applicable to DSCP trusted ports and untrusted ports. CoS rewrite is disabled on CoS trusted ports.

Example

- This command assigns the CoS of two to traffic classes 1, 3, and 5.

```
switch(config)#qos map traffic-class 1 3 5 to cos 2
switch(config)#
```

16.2.3 Transmit Queues and Port Shaping

A switch defines one traffic map-transmit queue map that applies to all Ethernet interfaces. Port shaping and transmit queues are configured on each Ethernet interface in interface ethernet configuration mode for the specified interface. Parameters for individual transmit queues are configured in tx-queue configuration command mode, which is entered from interface ethernet configuration mode.

The **tx-queue** command places the switch in Tx-queue configuration mode to configure a transmit queue on the configuration mode interface.

Example

- This command enters tx-queue configuration mode for transmit queue 3 of Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#tx-queue 3
switch(config-if-Et5-txq-3)#
```

Mapping Traffic Classes to a Transmit Queue

The **qos map traffic-class to tx-queue** command associates a transmit queue to a set of traffic classes. Multiple commands create a complete map, which the switch uses to schedule outbound traffic. A **show qos maps** command displays the traffic class to transmit queue map.

Example

- These commands assign traffic classes of 1, 3, and 5 to transmit queue 1, traffic classes 2, 4, and 6 to transmit queue 2, and traffic class 0 to transmit queue 0, then display the resultant map.

```
switch(config)#qos map traffic-class 1 3 5 to tx-queue 1
switch(config)#qos map traffic-class 2 4 6 to tx-queue 2
switch(config)#qos map traffic-class 0 to tx-queue 0
switch(config)#show qos maps
   Number of Traffic Classes supported: 7
   Number of Transmit Queues supported: 7

<-------OUTPUT OMITTED FROM EXAMPLE-------->

   Tc-queue map:
     tc:        0  1  2  3  4  5  6
     ------------------------------
     tx-queue:  0  1  2  1  2  1  2

switch(config)#
```

Configuring the Shape Rate – Port and Transmit Queues

A port's shape rate specifies the port's maximum outbound traffic bandwidth. A shape rate can also be configured for a port's transmit queue.

- To configure a port's shape rate, enter the interface configuration mode for the port and execute the **shape rate (Interface configuration mode)** command.
- To configure a transmit queue's shape rate, enter the transmit queue configuration mode for the queue and execute the **shape rate (Transmit Queue configuration modes)** command.

  Both shape rate commands use kbps to specify data rates.

Example

- These commands configure a shape rate of 5 Gbs on Ethernet port 3, then configure the shape rate for the following transmit queues:
  - — transmit queues 0, 1, and 2: 500 Mbps
  - — transmit queues 3, 4, and 5: 400 Mbps

```
switch(config)#interface ethernet 3
switch(config-if-Et3)#shape rate 5000000
switch(config-if-Et3)#tx-queue 0
switch(config-if-Et3-txq-0)#shape-rate 500000
switch(config-if-Et3-txq-0)#tx-queue 1
switch(config-if-Et3-txq-1)#shape rate 500000
switch(config-if-Et3-txq-1)#tx-queue 2
switch(config-if-Et3-txq-2)#shape rate 500000
switch(config-if-Et3-txq-2)#tx-queue 3
switch(config-if-Et3-txq-3)#shape rate 400000
switch(config-if-Et3-txq-3)#tx-queue 4
switch(config-if-Et3-txq-4)#shape rate 400000
switch(config-if-Et3-txq-4)#tx-queue 5
switch(config-if-Et3-txq-5)#shape rate 400000
switch(config-if-Et3-txq-5)#exit
```

The **show qos interface** displays the shape rate configuration for the specified port.

```
switch(config-if-Et3-txq-5)#show qos interface ethernet 3
Ethernet3:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 5000000Kbps

   Tx-Queue    Bandwidth     ShapeRate      Priority
               (percent)        (Kbps)
   ----------------------------------------------
          0          N/A       500000        strict
          1          N/A       500000        strict
          2          N/A       500000        strict
          3          N/A       400000        strict
          4          N/A       400000        strict
          5          N/A       400000        strict
          6          N/A     disabled        strict
switch(config-if-Et3-txq-5)#
```

Configuring Queue Priority

The **priority (Trident platform switches)** command configures a transmit queue's priority type:

- The **priority strict** command configures the queue as a strict priority queue.
- The **no priority** command configures the queue as a round robin queue.

  A queue's configuration as ***round robin*** also applies to all lower priority queues regardless of other configuration statements.

The **bandwidth percent (Transmit Queue Configuration Modes)** command configures a round robin queue's bandwidth share. The cumulative allocated bandwidth of all round robin queues is always 100%. If the cumulative configured bandwidth is less than 100%, the remaining bandwidth is

distributed equally to the queues. If the cumulative configured bandwidth is greater than 100%, each port's allocated bandwidth is its configured bandwidth divided by the cumulative configured bandwidth.

Example

- These commands configure tranmsit queue 3 as a round robin queue, then allocates 10%, 20%, 30%, and 40% bandwidth to queues 0 through 3. The **priority** statement for queue 3 also configures priority for queues 0, 1, and 2.

  Removing the statement reverts the other queues to ***strict priority*** type unless ***running-config*** contains a **no priority** statement for one of these queues.

```
switch(config-if-Et3)#tx-queue 3
switch(config-if-Et3-txq-3)#no priority
switch(config-if-Et3-txq-3)#bandwidth percent 40
switch(config-if-Et3-txq-3)#tx-queue 2
switch(config-if-Et3-txq-2)#bandwidth percent 30
switch(config-if-Et3-txq-2)#tx-queue 1
switch(config-if-Et3-txq-1)#bandwidth percent 20
switch(config-if-Et3-txq-1)#tx-queue 0
switch(config-if-Et3-txq-0)#bandwidth percent 10
```

  The **show qos interface** displays the priority type and bandwidth for each transmit queue.

```
switch(config-if-Et3-txq-0)#show qos interface ethernet 3
Ethernet3:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 5000000Kbps

   Tx-Queue    Bandwidth     ShapeRate       Priority
               (percent)        (Kbps)
   ------------------------------------------------
          0           10       500000   round-robin
          1           20       500000   round-robin
          2           30       500000   round-robin
          3           40       400000   round-robin
          4          N/A       400000        strict
          5          N/A       400000        strict
          6          N/A     disabled        strict

switch(config-if-Et3-txq-0)#
```

  Changing the configured bandwidth percentage for queue 3 from 40 to 12 adds 7% ((40-12)/4)to the bandwidth of each queue.

```
Tx-Queue    Bandwidth     ShapeRate       Priority
               (percent)        (Kbps)
   ------------------------------------------------
          0           17       500000   round-robin
          1           27       500000   round-robin
          2           37       500000   round-robin
          3           19       400000   round-robin
          4          N/A       400000        strict
          5          N/A       400000        strict
          6          N/A     disabled        strict
```

Changing the configured bandwidth percentage for queue 3 to 60 changes the allocated bandwidth of each queue to its allocated bandwidth divided by 120% (10%+20%+30%+60%).

```
Tx-Queue    Bandwidth     ShapeRate       Priority
               (percent)        (Kbps)
   ------------------------------------------------
          0           8       500000   round-robin
          1          16       500000   round-robin
          2          25       500000   round-robin
          3          50       400000   round-robin
          4         N/A       400000        strict
          5         N/A       400000        strict
          6         N/A     disabled        strict
```

16.3 Quality of Service Configuration Commands

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Commands

- platform petraA traffic-class . . . . . Page 656
- qos map cos . . . . . Page 664
- qos map dscp . . . . . Page 665
- qos map traffic-class to cos . . . . . Page 666
- qos map traffic-class to dscp . . . . . Page 667
- qos map traffic-class to mc-tx-queue . . . . . Page 668
- qos map traffic-class to tx-queue . . . . . Page 670
- qos map traffic-class to uc-tx-queue . . . . . Page 671
- qos rewrite cos . . . . . Page 673
- qos rewrite dscp . . . . . Page 674

Interface Configuration Commands

- mc-tx-queue . . . . . Page 655
- qos cos . . . . . Page 661
- qos dscp . . . . . Page 662
- qos trust . . . . . Page 663
- shape rate (Interface configuration mode) . . . . . Page 675
- tx-queue . . . . . Page 679
- uc-tx-queue . . . . . Page 680

EXEC Commands

- show qos interface . . . . . Page 677
- show qos maps . . . . . Page 678

Tx Queue Configuration Commands

- bandwidth percent (Transmit Queue Configuration Modes) . . . . . Page 653
- priority (FM4000 and Petra platform switches) . . . . . Page 657
- priority (Trident platform switches) . . . . . Page 659
- shape rate (Transmit Queue configuration modes) . . . . . Page 676

bandwidth percent (Transmit Queue Configuration Modes)

The **bandwidth percent** command configures the bandwidth share of the configuration mode transmit queue when it is configured as a round robin queue. When the cumulative configured bandwidth of all round robin queues is less than 100%, the remaining bandwidth is shared equally by all queues. When the cumulative configured bandwidth is greater than 100%, each queue's share is adjusted to provide a bandwidth proportional with the other queues' share. The default value is 0%.

The **no bandwidth percent** and **default bandwidth percent** commands restore the default bandwidth share of the configuration mode transmit queue by removing the corresponding from ***running-config***.

Command Mode

Tx-Queue Configuration

Mc-Tx-Queue configuration (Trident platform switches)
Uc-Tx-Queue configuration (Trident platform switches)

Command Syntax

```
bandwidth percent proportion
no bandwidth percent
default bandwidth percent
```

Parameters

- *proportion* Bandwidth percentage assigned to queues. Values range from 1 to 100.

Guidelines

Trident platform switches allocate unused bandwidth among all unicast and multicast queues.

Example

- These commands configure the bandwidth share of three queues at 30% and one queue at 10%.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#tx-queue 0
switch(config-if-Et2-txq-0)#bandwidth percent 30
switch(config-if-Et2-txq-0)#tx-queue 1
switch(config-if-Et2-txq-1)#bandwidth percent 30
switch(config-if-Et2-txq-1)#tx-queue 2
switch(config-if-Et2-txq-2)#bandwidth percent 30
switch(config-if-Et2-txq-2)#tx-queue 3
switch(config-if-Et2-txq-3)#bandwidth percent 10
switch(config-if-Et2-txq-3)#show qos interface ethernet 2
Ethernet2:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 1000000Kbps
```

```
   Tx-Queue   Bandwidth    ShapeRate      Priority
              (percent)       (Kbps)
   ------------------------------------------------
          0          30      1000000   round-robin
          1          30      1000000   round-robin
          2          30      1000000   round-robin
          3          10      1000000   round-robin
          4         N/A      1000000        strict
          5         N/A      1000000        strict
          6         N/A      1000000        strict
```

- These commands re-configures the bandwidth share of the fourth queue at 30%.

```
switch(config-if-Et2-txq-3)#tx-queue 3
switch(config-if-Et2-txq-3)#bandwidth percent 30
switch(config-if-Et2-txq-3)#show qos interface ethernet 2
Ethernet2:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 1000000Kbps

   Tx-Queue   Bandwidth    ShapeRate      Priority
              (percent)       (Kbps)
   ------------------------------------------------
          0          25      1000000   round-robin
          1          25      1000000   round-robin
          2          25      1000000   round-robin
          3          25      1000000   round-robin
          4         N/A      1000000        strict
          5         N/A      1000000        strict
          6         N/A      1000000        strict
```

- These commands re-configures the bandwidth share of the fourth queue at 2%.

```
switch(config-if-Et2-txq-3)#tx-queue 3
switch(config-if-Et2-txq-3)#bandwidth percent 2
switch(config-if-Et2-txq-3)#show qos interface ethernet 2
Ethernet2:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 1000000Kbps

   Tx-Queue   Bandwidth    ShapeRate      Priority
              (percent)       (Kbps)
   ------------------------------------------------
          0          32      1000000   round-robin
          1          32      1000000   round-robin
          2          32      1000000   round-robin
          3           4      1000000   round-robin
          4         N/A      1000000        strict
          5         N/A      1000000        strict
          6         N/A      1000000        strict

switch(config-if-Et2-txq-3)#
```

mc-tx-queue

The **mc-tx-queue** command places the switch in mc-tx-queue configuration mode to configure a multicast transmit queue on the configuration mode interface. Mc-tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Trident switches have four multicast queues (MC0 – MC03) and eight unicast queues (UC0 – UC7), categorized into two priority groups. All queues are exposed through the CLI and are user configurable.

- Priority Group 1: UC7, UC6, MC3
- Priority Group 0: UC5, UC4, MC2, UC3, UC2, MC1, UC1, UC0, MC0

The **exit** command returns the switch to the originating interface configuration mode.

Command Mode

Interface-Ethernet configuration

Command Syntax

mc-tx-queue *queue_level*

Parameters

- *queue_level* The multicast transmit queue number. Values range from ***0*** to ***3***.

Commands Available in tx-queue configuration mode:

- **bandwidth percent (Transmit Queue Configuration Modes)**
- **priority (Trident platform switches)**
- **shape rate (Transmit Queue configuration modes)**

Restrictions

This command is available on Trident platform switches.

Related Commands

- **uc-tx-queue**: Configures unicast transmit queues on Trident platform switches.
- **tx-queue**: Configures transmit queues on FM4000 and Trident platform switches.

Example

- This command enters mc-tx-queue configuration mode for multicast transmit queue 3 of Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#mc-tx-queue 3
switch(config-if-Et5-mc-txq-3)#
```

platform petraA traffic-class

The **platform petraA traffic-class** command specifies the default traffic class used by all ports on a specified chip. The default traffic class is an alternative QoS and DSCP configuration that Petra switches implement, replacing **qos cos** and **qos dscp** commands. This command is valid only on Petra switches.

Traffic class values range from 0 to 6. The default traffic class value depends on the switch model.

When **platform ?** returns *Petra*:

- CoS trusted ports: inbound untagged packets are assigned to the default traffic class. Tagged packets are assigned to the traffic class that corresponds to the contents of its CoS field.
- DSCP trusted ports: inbound non-IP packets are assigned to the default traffic class. IP packets are assigned to the traffic class that corresponds to the contents of its DSCP field.
- Untrusted ports: all inbound packets are assigned to the default traffic class.

The **no platform petraA traffic-class** and **default platform petraA traffic-class** commands restore the default traffic class of one by deleting the corresponding **platform petraA traffic-class** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
platform petraA [CHIP_NAME] traffic-class tc_value
no platform petraA traffic-class
default platform petraA traffic-class
```

Parameters

- *CHIP_NAME* trust mode assigned to the specified ports. Port designation options include:
  - <no parameter> all ports on the switch.
  - **module** *card_x* all ports on linecard specified by *card_x*.
  - **linecard***card_x***-petra-***chip_y* all ports on PetraA chip *chip_y* on linecard *card_x*.

  Each PetraA switch can contain up to ten linecards. Values of *card_x* vary from 3 to 10.

  PetraA chips on each linecard control eight ports. Values of *chip_y* vary from 0 to 5:

  - 0 controls ports 1 through 8
  - 1 controls ports 9 through 16
  - 2 controls ports 17 through 24
  - 3 controls ports 25 through 32
  - 4 controls ports 33 through 40
  - 5 controls ports 41 through 48
- *tc_value* Traffic class value. Values range from 0 to 7. Default value is 1.

Restrictions

This command is available on Petra platform switches.

Example

- This command configures the default traffic class to five for the ports 25-32 on linecard 5.

```
switch(config)#platform petraA linecard5-Petra-3 traffic-class 5
switch(config)#
```

priority (FM4000 and Petra platform switches)

The **priority** command specifies the priority of the configuration mode transmit queue. The switch supports two queue priorities:

- strict priority: contents are removed from the queue, subject to maximum bandwidth limits, before data from lower priority queues. Tx-queue 7 is set to strict priority and is not configurable. The default setting on all queues is strict priority.
- round robin priority: contents are removed proportionately from all round robin queues, subject to maximum bandwidth limits assigned to the strict priority queues.

When a queue is configured as a round robin queue, all lower priority queues also function as round robin queues. A queue's priority rank is indicated by its numerical label: higher labels denote higher priority. Tx-queue 6 has higher priority than Tx-queue 5, and Tx-queue 0 has the lowest priority.

The **priority strict** and **default priority** commands configure a transmit queue to function as a strict priority queue unless a higher priority queue is configured as a round robin queue.

The **no priority** command configures a transmit queue as a round robin queue. All lower priority queues also function as round robin queues regardless of their configuration.

Command Mode

Tx-Queue Configuration

Command Syntax

```
priority strict
no priority
default priority
```

Related Commands

- **tx-queue** places the switch in tx-queue configuration mode.

Example

- The first command displays the default state of all transmit queues on Ethernet interface 2. The second command configures transmit queue 3 as a round robin queue. The third command displays the effect of the no priority command on all transmit queues on the interface.

```
switch(config)#interface ethernet 2
switch(config-if-Et2)#show qos interface ethernet 2
Ethernet2:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 1000000Kbps

   Tx-Queue   Bandwidth    ShapeRate      Priority
              (percent)       (Kbps)
   ------------------------------------------------
          0         N/A      1000000        strict
          1         N/A      1000000        strict
          2         N/A      1000000        strict
          3         N/A      1000000        strict
          4         N/A      1000000        strict
          5         N/A      1000000        strict
          6         N/A      1000000        strict
```

```
switch(config-if-Et2)#tx-queue 3
switch(config-if-Et2-txq-3)#no priority
switch(config-if-Et2-txq-3)#show qos interface ethernet 2
Ethernet2:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 1000000Kbps

   Tx-Queue   Bandwidth    ShapeRate      Priority
              (percent)       (Kbps)
   ------------------------------------------------
          0          32      1000000   round-robin
          1           2      1000000   round-robin
          2          22      1000000   round-robin
          3          42      1000000   round-robin
          4         N/A      1000000        strict
          5         N/A      1000000        strict
          6         N/A      1000000        strict

switch(config-if-Et2-txq-3)#
```

priority (Trident platform switches)

The **priority** command specifies the priority of the configuration mode transmit queue. The switch supports two queue priorities:

- strict priority: contents are removed from the queue, subject to maximum bandwidth limits, before data from lower priority queues. The default setting on all other queues is strict priority.
- round robin priority: contents are removed proportionately from all round robin queues, subject to maximum bandwidth limits assigned to the strict priority queues.

Trident switches have eight unicast queues (UC0 – UC7) and four multicast queues (MC0 – MC03), categorized into two priority groups. Priority group 1 queues have priority over priority 0 queues. The following lists display the priority group queues in order from higher priority to lower priority.

- Priority Group 1: UC7, UC6, MC3
- Priority Group 0: UC5, UC4, MC2, UC3, UC2, MC1, UC1, UC0, MC0

Priority group 1 queues are strict priority queues and are not configurable as round robin. Priority 0 queues are strict priority by default and are configurable as round robin. When a queue is configured as a round robin queue, all lower priority queues automatically function as round robin queues.

The **priority strict** and **default priority** commands configure a transmit queue to function as a strict priority queue unless a higher priority queue is configured as a round robin queue.

The **no priority** command configures a transmit queue as a round robin queue. All lower priority queues also function as round robin queues regardless of their configuration.

Command Mode

mc-tx-Queue configuration
uc-tx-Queue configuration

Command Syntax

```
priority strict
no priority
default priority
```

Related Commands

- **mc-tx-queue** places the switch in mc-tx-queue configuration mode.
- **uc-tx-queue**: places the switch in uc-tx-queue configuration mode.
- **priority (FM4000 and Petra platform switches)** is the equivalent command on FM4000 and Petra platform switches.

Example

- The first command displays the default state of all transmit queues on Ethernet interface 10. The second command configures unicast transmit queue 3 as a round robin queue. The third command displays the effect of the no priority command on all transmit queues on the interface.

```
switch(config)#interface ethernet 10
switch(config-if-Et10)#show qos interface ethernet 10
Ethernet10:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled
```

```
   Tx-Queue   Bandwidth    Shape Rate     Priority   Priority Group
              (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A     disabled        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A     disabled        strict                0
        UC3         N/A     disabled        strict                0
        UC2         N/A     disabled        strict                0
        MC1         N/A     disabled        strict                0
        UC1         N/A     disabled        strict                0
        UC0         N/A     disabled        strict                0
        MC0         N/A     disabled        strict                0

switch(config-if-Et10)#uc-tx-queue 3
switch(config-if-Et10-uc-txq-3)#no priority
switch(config-if-Et10-uc-txq-3)#show qos interface ethernet 10
Ethernet10:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: disabled

   Tx-Queue   Bandwidth    Shape Rate     Priority   Priority Group
              (percent)       (Kbps)
   ------------------------------------------------------------------
        UC7         N/A     disabled        strict                1
        UC6         N/A     disabled        strict                1
        MC3         N/A     disabled        strict                1
        UC5         N/A     disabled        strict                0
        UC4         N/A     disabled        strict                0
        MC2         N/A     disabled        strict                0
        UC3          16     disabled   round-robin                0
        UC2          16     disabled   round-robin                0
        MC1          16     disabled   round-robin                0
        UC1          16     disabled   round-robin                0
        UC0          16     disabled   round-robin                0
        MC0          16     disabled   round-robin                0

switch(config-if-Et10-uc-txq-3)#
```

qos cos

The **qos cos** command specifies the default class of service (CoS) value of the configuration mode interface. CoS values range from 0 to 7. Default value is 0.

When **platform ?** returns ***fm4000*** or ***trident***:

- CoS trusted ports: the default CoS value determines the traffic class for inbound untagged packets. Tagged packets are assigned to the traffic class that corresponds to the contents of its CoS field.
- Untrusted ports: the default CoS value determines the traffic class for all inbound packets.

When **platform ?** returns ***PetraA***:

- CoS trusted ports: inbound untagged packets are assigned to the default traffic class, as configured by the **platform petraA traffic-class** command. Tagged packets are assigned to the traffic class that corresponds to the contents of its CoS field.
- Untrusted ports: all inbound packets are assigned to the default traffic class.

The **no qos cos** and **default qos cos** commands restore the port's default CoS value to zero by deleting the corresponding **qos cos** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
qos cos cos_value
no qos cos
default qos cos
```

Parameters

- *cos_value* CoS value assigned to port. Value ranges from 0 to 7. Default value is 0.

Restrictions

The **qos cos** command has no effect on Petra platform switches.

Example

- This command configures the default CoS of four on Ethernet interface 8.

```
switch(config-if-Et8)#qos cos 4
switch(config-if-Et8)#
```

qos dscp

The **qos dscp** command specifies the default differentiated services code point (DSCP) value of the configuration mode interface. The default DSCP determines the traffic class for non-IP packets that are inbound on DSCP trusted ports. DSCP trusted ports determine the traffic class for inbound packets as follows:

- **platform ?** returns ***fm4000*** or ***trident***:
  - non-IP packets: default DSCP value specified by **qos dscp** determines the traffic class.
  - IP packets: assigned to the traffic class corresponding to its DSCP field contents.
- **platform ?** returns ***PetraA***:
  - non-IP packets: assigned to the default traffic class configured by **platform petraA traffic-class**.
  - IP packets: assigned to the traffic class corresponding to its DSCP field contents.

The **no qos dscp** and **default qos dscp** commands restore the port's default DSCP value to zero by deleting the corresponding **qos dscp** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
qos dscp dscp_value
no qos dscp
default qos dscp
```

Parameters

- *dscp_value* DSCP value assigned to the port. Value ranges from 0 to 63. Default value is 0.

Restrictions

The **qos dscp** command has no effect on Petra platform switches.

Example

- This command sets the default DSCP of 44 on Ethernet 7 interface.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#qos dscp 44
switch(config-if-Et7)
```

qos trust

The **qos trust** command configures the quality of service port trust mode for the configuration mode interface. Trust-enabled ports classify traffic by examining the traffic's CoS or DSCP value. Port trust state default setting is ***cos***.

The **no qos trust** command places the port in ***untrusted*** mode.

The **default qos trust** command restores the default trust mode of ***cos*** on the configuration mode interface by removing the corresponding **qos trust** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port Channel Configuration

Command Syntax

```
qos trust MODE
no qos trust
default qos trust
```

Parameters

- *MODE* trust mode assigned to the port. Options include:
  - **cos** enables cos trust mode.
  - **dscp** enables dscp trust mode.

  no qos trust enables untrusted mode on the port.

Examples

- This command configures trust mode of dscp for Ethernet interface 5.

```
switch(config)#interface Ethernet 7
switch(config-if-Et7)#qos trust dscp
switch(config-if-Et7)#
```

- This command configures trust mode of untrusted for Port Channel interface 23.

```
switch(config)#interface port-channel 23
switch(config-if-Po23)#no qos trust
switch(config-if-Po23)#
```

qos map cos

The **qos map cos** command associates a traffic class to a list of Class of Service (CoS) settings. Multiple commands create a complete CoS to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's CoS field or the port upon which it is received.

The **no qos map cos** and **default qos map cos** commands restore the specified CoS values to their default traffic class setting by deleting the corresponding **qos map cos** statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos map cos cos_num_1 [cos_num_2 ... cos_num_n] to traffic-class tc_value
no qos map cos cos_num_1 [cos_num_2 ... cos_num_n]
default qos map cos cos_num_1 [cos_num_2 ... cos_num_n]
```

Parameters

- *cos_value_x* Class of Service (CoS) value. Values range from 0 to 7.
- *tc_value* Traffic class value. Value range varies by platform.

Default CoS to traffic class map varies by platform (Table 16-18).

Default Inbound CoS to TC Map

Table 16-18 displays the default Class of Service to traffic class map for each platform.

Inbound CoS	Traffic Class		
	fm4000	Petra	Trident
untagged	1	1	1
0	1	1	1
1	0	0	0
2	2	2	2
3	3	3	3
4	4	4	4
5	4	5	5
6	5	6	6
7	6	7	7

Table 16-18 Cos to Traffic Class Map

Example

- This command assigns the traffic class of 5 to the classes of service 1, 3, 5, and 7.

```
switch(config)#qos map cos 1 3 5 7 to traffic-class 5
switch(config)#
```

qos map dscp

The **qos map dscp** command associates a traffic class to a set of differentiated services control point (DSCP) values. Multiple commands create a complete DSCP to traffic class map. The switch uses this map to assign a traffic class to data packets on the basis of the packet's DSCP field or the port upon which it is received.

The **no qos map dscp** and **default qos map dscp** commands restore the specified DSCP values to their default traffic class settings by deleting corresponding **qos map dscp** statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos map dscp dscp_v_1 [dscp_v_2 ... dscp_v_n] to traffic-class tc_value
no qos map dscp dscp_v_1 [dscp_v_2 ... dscp_v_n]
default qos map dscp dscp_v_1 [dscp_v_2 ... dscp_v_n]
```

Parameters

- *dscp_v_x* Differentiated services control point (DSCP) value. Values range from 0 to 63.
- *tc_value* Traffic class value. Value range varies by platform,

  Default map varies by platform (Table 16-19).

Default Inbound DSCP to TC Map

Table 16-19 displays the default DSCP to traffic class map for each platform.

Inbound DSCP	Traffic Class		
	fm4000	Petra	Trident
0-7	1	1	1
8-15	0	0	0
16-23	2	2	2
24-31	3	3	3
32-39	4	4	4
40-47	4	5	5
48-55	5	6	6
56-63	5	7	7

Table 16-19 DSCP to Traffic Class Map

Examples

- This command assigns the traffic class of three to the DSCP values of 12, 13, 25, and 37.

```
switch(config)#qos map dscp 12 13 25 37 to traffic-class 3
switch(config)#
```

qos map traffic-class to cos

The **qos map traffic-class to cos** command associates a Class of Service (CoS) to a list of traffic classes. Multiple commands create a complete traffic class to CoS map. The switch uses this map to fill the CoS field in outbound packets. This map is applicable to DSCP trusted ports and untrusted ports. CoS rewrite is disabled on CoS trusted ports. The **show qos maps** command displays the CoS to traffic class map.

The **no qos traffic-class to cos** and **default qos traffic-class to cos** commands restore the specified traffic class values to their default CoS settings by removing the corresponding **qos map traffic-class to cos** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to cos cos_value
no qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to cos
default qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to cos
```

Parameters

- *tc_num_x* Traffic class value. Values vary with switch platform.
- *cos_value* Class of Service (CoS) value. Values range from 0 to 7.

Default Inbound Traffic Class to CoS Map

Table 16-20 displays the default traffic class to Class of Service map for each platform.

Traffic Class	CoS		
	fm4000	Petra	Trident
0	1	1	1
1	0	0	0
2	2	2	2
3	3	3	3
4	4	4	4
5	6	5	5
6	7	6	6
7	X	7	7

Table 16-20 Default Traffic Class to CoS Map

Example

- This command assigns the CoS of two to traffic classes 1, 3, and 5.

```
switch(config)#qos map traffic-class 1 3 5 to cos 2
switch(config)#
```

qos map traffic-class to dscp

The **qos map traffic-class to dscp** command associates a differentiated services control point (DSCP) value to a list of traffic classes. Multiple commands create a complete traffic class to DSCP map. The switch uses this map in DSCP rewrite operations to fill the DSCP field in outbound packets. This map is applicable to CoS trusted ports and untrusted ports but disabled by default on these ports. DSCP rewrite is disabled on DSCP trusted ports. The **show qos maps** command displays the traffic class to DSCP map.

The **no qos traffic-class to dscp** and **default qos traffic-class to dscp** commands restore the specified traffic class values to their default DSCP settings by removing the corresponding **qos map traffic-class to dscp** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to dscp dscp_value
no qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to dscp
default qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to dscp
```

Parameters

- *tc_num_x* Traffic class value. Values vary with switch platform.
- *dscp_value* Differentiated services control point (DSCP) value. Values range from 0 to 63.

Default Inbound Traffic Class to DSCP Map

Table 16-21 displays the default traffic class to DSCP map for each platform.

Traffic Class	DSCP		
	fm4000	Petra	Trident
0	8	NA	8
1	0	NA	0
2	16	NA	16
3	24	NA	24
4	32	NA	32
5	48	NA	40
6	56	NA	48
7	X	NA	56

Table 16-21 Default Traffic Class to DSCP Map

Restrictions

This command is not available on Petra platform switches.

Example

- This command assigns the DSCP value of 17 to traffic classes 1, 2, and 4.

```
switch(config)#qos map traffic-class 1 2 4 to dscp 17
switch(config)#
```

qos map traffic-class to mc-tx-queue

The **qos map traffic-class to mc-tx-queue** command associates a multicast transmit queue to a list of traffic classes. Multiple commands create a complete traffic class to mc-tx-queue map. The switch uses this map to route outbound packets to transmit queues, which in turn schedules their transmission from the switch.The **show qos maps** command displays the traffic class to multicast transmit queue map.

The **no qos traffic-class to mc-tx-queue** and **default qos traffic-class to mc-tx-queue** commands restore the default traffic class to multicast transmit queue map for the specified traffic class values by removing the corresponding **qos map traffic-class to mc-tx-queue** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to mc-tx-queue mtq_value
no qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to mc-tx-queue
default qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to mc-tx-queue
```

Parameters

- *tc_num_x* Traffic class value. Values range from 0 to 7.
- *mtq_value* Multicast transmit queue number. Values range from ***0*** to ***3***.

Restrictions

This command is available on Trident platform switches. The **qos map traffic-class to tx-queue** command configures transmit queues on FM4000 and Petra platform switches.

Default Inbound Traffic Class to Tx-Queue Map

Table 16-22 displays the default traffic class to multicast transmit queue map.

Traffic Class	Transmit Queue		
	fm4000	Petra	Trident
0	NA	NA	0
1	NA	NA	0
2	NA	NA	1
3	NA	NA	1
4	NA	NA	2
5	NA	NA	2
6	NA	NA	3
7	NA	NA	3

Table 16-22 Default Traffic Class to Multicast Transmit Queue Map

Related Commands

The **qos map traffic-class to uc-tx-queue** command configures the traffic class to unicast transmit queue map.

Example

- This command maps traffic classes 0, 4, and 5 to mc-tx-queue 2.

```
switch(config)#qos map traffic-class 0 4 5 to mc-tx-queue 2
switch(config)#
```

qos map traffic-class to tx-queue

The **qos map traffic-class to tx-queue** command associates a transmit queue (tx-queue) to a list of traffic classes. Multiple commands create a complete traffic to tx-queue map. The switch uses this map to route outbound packets to transmit queues, which in turn schedules their transmission from the switch.The **show qos maps** command displays the transmit queue to traffic class map.

The **no qos traffic-class to tx-queue** and **default qos traffic-class to tx-queue** commands restore the specified traffic class values to their default transmit queue settings by removing the corresponding **qos map traffic-class to tx-queue** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to tx-queue txq_value
no qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to tx-queue
default qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to tx-queue
```

Parameters

- *tc_num_x* Traffic class value. Values range from 0 to 7. Default varies with platform and *cos_value*.
- *txq_value* Transmit queue value.

Restrictions

This command is available on FM4000 and Petra platform switches. The **qos map traffic-class to mc-tx-queue** (multicast queues) and **qos map traffic-class to uc-tx-queue** (unicast queues) commands configure traffic class to transmit queue maps on Trident platform switches.

FM4000 platform switches: When priority flow control (PFC) is enabled, the traffic classes are mapped 1-1 to the transmit queue, regardless of existing **qos map traffic-class to tx-queue** commands.

Default Inbound Traffic Class to Tx-Queue Map

The transmit queue to traffic class map varies by platform, as shown by Table 16-23.

Traffic Class	Transmit Queue		
	fm4000	Petra	Trident
0	0	0	NA
1	1	1	NA
2	2	2	NA
3	3	3	NA
4	4	4	NA
5	5	5	NA
6	6	6	NA
7	X	7	NA

Table 16-23 Default Traffic Class to Transmit Queue Map

Example

- This command maps traffic classes 0, 4, and 5 to tx-queue 4.

```
switch(config)#qos map traffic-class 0 4 5 to tx-queue 4
```

qos map traffic-class to uc-tx-queue

The **qos map traffic-class to uc-tx-queue** command associates a unicast transmit queue to a list of traffic classes. Multiple commands create a complete traffic class to unicast transmit queue map. The switch uses this map to route outbound packets to transmit queues, which in turn schedules their transmission from the switch.The **show qos maps** command displays the traffic class to unicast transmit queue map.

The **no qos traffic-class to uc-tx-queue** and **default qos traffic-class to uc-tx-queue** commands restore the default traffic class to unicast transmit queue map for the specified traffic class values by removing the corresponding **qos map traffic-class to uc-tx-queue** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to uc-tx-queue utq_value
no qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to uc-tx-queue
default qos map traffic-class tc_num_1 [tc_num_2 ... tc_num_n] to uc-tx-queue
```

Parameters

- *tc_num_x* Traffic class value. Values range from 0 to 7.
- *utq_value* Unicast transmit queue number. Value ranges from 0 to 7.

Restrictions

This command is available on Trident platform switches. The **qos map traffic-class to tx-queue** command configures transmit queues on FM4000 and Petra platform switches.

Default Inbound Traffic Class to Tx-Queue Map

Table 16-24 displays the default traffic class to multicast transmit queue map.

Traffic Class	Transmit Queue		
	fm4000	Petra	Trident
0	NA	NA	0
1	NA	NA	1
2	NA	NA	2
3	NA	NA	3
4	NA	NA	4
5	NA	NA	5
6	NA	NA	6
7	NA	NA	7

Table 16-24 Default Traffic Class to Unicast Transmit Queue Map

Related Commands

The **qos map traffic-class to mc-tx-queue** command configures the traffic class to unicast transmit queue map.

Example

- This command maps traffic classes 0, 4, and 5 to unicast transmit queue 4.

```
switch(config)#qos map traffic-class 0 4 5 to uc-tx-queue 4
switch(config)#
```

qos rewrite cos

The **qos rewrite cos** command enables the rewriting of the CoS field for outbound tagged packets that are received on DSCP trusted ports and untrusted ports. CoS rewrite is always disabled on CoS trusted ports. The CoS value that is written into the packet is configurable and based on the data stream's traffic class. CoS rewriting is active by default.

The **no qos rewrite cos** command disables CoS rewriting on the switch. The default **qos rewrite cos** command restores the default setting of enabling CoS rewriting by removing the **no qos rewrite cos** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos rewrite cos
no qos rewrite cos
default qos rewrite cos
```

Restrictions

The **qos rewrite cos** command is not available on Petra platform switches.

Related Commands

- **qos map traffic-class to cos**: Configures the traffic class to CoS rewrite map.

Example

- This command enables CoS rewrite.

```
switch(config)#qos rewrite cos
switch(config)#
```

qos rewrite dscp

The **qos rewrite dscp** command enables the rewriting of the DSCP field for outbound tagged packets that are received on CoS trusted ports and untrusted ports. DSCP rewrite is always disabled on DSCP trusted ports. The DSCP value that is written into the packet is configurable and based on the data stream's traffic class. DSCP rewriting is disabled by default and not available on Petra platform switches.

The **no qos rewrite dscp** and **default qos rewrite dscp** commands disable DSCP rewriting on the switch by removing the **no qos rewrite dscp** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
qos rewrite dscp
no qos rewrite dscp
default qos rewrite dscp
```

Restrictions

The **qos rewrite dscp** command is not available on Petra platform switches.

Related Commands

- **qos map traffic-class to dscp**: Configures the traffic class to DSCP rewrite map.

Example

- This command enables DSCP rewrite.

```
switch(config)#qos rewrite dscp
switch(config)#
```

shape rate (Interface configuration mode)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode interface, also known as queue shaping. The shape rate for individual transmit queues is configured by the **shape rate (Transmit Queue configuration modes)** command. By default, outbound transmission rate is not bounded by a shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode interface by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration

Command Syntax

```
shape rate byte_limit
no shape rate
default shape rate
```

Parameters

- *byte_limit* shape rate applied to interface (Kbps). Valid options vary by interface type.

Guidelines – FM4000 Platform Switches

Shaping rates greater than 460 kbps are supported. At shaping rates smaller than 10 Mbps, granularity and rounding errors may skew the actual shaping rate by 20% from the specified rate.

Enabling port shaping on an interface disables queue shaping internally. Disabling port shaping restores queue shaping as specified in ***running-config***.

Guidelines – Trident Platform Switches

Shaping rates of at least 8 kbps are supported. At shaping rates smaller than 1 Mbps, granularity and rounding errors may skew the actual shaping rate by 20% from the specified rate.

Guidelines – Petra Platform Switches

The following port shaping rates are supported:

- 1G ports: above 100 kbps.
- 10G ports: above 7900 kbps.

Commands that specify a smaller shape rate disable port shaping on the interface.

Example

- This command configures a port shape rate of 5 Gbps on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#shape rate 5000000
switch(config-if-Et5)#
```

shape rate (Transmit Queue configuration modes)

The **shape rate** command specifies the maximum bandwidth for outbound traffic on the configuration mode transmit queue, also known as queue shaping. The shape rate for interfaces is configured by the **shape rate (Interface configuration mode)** command. By default, the configured outbound transmission rate is not bounded by a transmit queue shape rate.

The **no shape rate** and **default shape rate** commands remove the shape rate bandwidth limit on the configuration mode transmit queue by deleting the corresponding **shape rate** command from ***running-config***.

Command Mode

Tx-Queue configuration (FM4000 and Petra platform switches)
Mc-Tx-Queue configuration (Trident platform switches)
Uc-Tx-Queue configuration (Trident platform switches)

Command Syntax

```
shape rate byte_limit
no shape rate
default shape rate
```

Parameters

- *byte_limit* shape rate applied to interface (Kbps). Valid options vary by interface type.

Guidelines – FM4000 Platform Switches

Queue shaping on a port is supported only when port shaping is not enabled on the interface. Enabling port shaping on a port disables queue shaping internally. Disabling port shaping restores queue shaping as specified by ***running-config***.

Shaping rates greater than 460 kbps are supported. At lower shaping rates (less than 10 Mbps), granularity and rounding errors may skew the actual shaping rate by 20% from the specified rate.

Guidelines – Petra Platform Switches

Queue shaping applies only to unicast traffic.

Shaping rates of at least 162 Kbps are supported.

Guidelines – Trident Platform Switches

Shaping rates of at least 8 kbps are supported. At shaping rates smaller than 1 Mbps, granularity and rounding errors may skew the actual shaping rate by 20% from the specified rate.

When two queues source traffic from the same traffic class and the higher priority queue is shaped, the higher priority queue consumes all internal buffers, which starves the lower priority queue even when bandwidth is available to the port.

Example

- This command configures a shape rate of 1 Gbps (1,000,000 Kbps) on transmit queue 3 of Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#tx-queue 3
switch(config-if-Et5-txq-3)#shape rate 1000000
switch(config-if-Et5-txq-3)#
```

show qos interface

The **show qos interface** command displays the QoS, DSCP, and transmit queue configuration on a specified interface. Information provided by this command includes the ports trust setting, the default CoS value, and the DSCP value.

Configurable transmit queues are not supported on Trident platform switches.

Command Mode
EXEC

Command Syntax
show qos interface ***INTERFACE_NAME***

Parameters

- *INTERFACE_NAME* Interface For which command returns data. Options include:
  - <no parameter> returns data for all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.

Examples

- This command lists the QoS configuration for Ethernet interface 4.

```
switch>show qos interface ethernet 4
Ethernet4:
   Trust Mode: COS
   Default COS: 0
   Default DSCP: 0

   Port shaping rate: 5000000Kbps

   Tx-Queue   Bandwidth    ShapeRate      Priority
              (percent)       (Kbps)
   ------------------------------------------------
          0          50     disabled   round-robin
          1          50     disabled   round-robin
          2         N/A     disabled        strict
          3         N/A      1000000        strict
          4         N/A      1000000        strict
          5         N/A      1500000        strict
          6         N/A      2000000        strict

switch>
```

show qos maps

The **show qos maps** command lists the number of traffic classes that the switch supports and displays the CoS-Traffic Class, DSCP-Traffic Class, Traffic Class-CoS, and Traffic Class-Transmit Queue maps.

Command Mode

EXEC

Command Syntax

show qos maps

Examples

- This command displays the QoS maps that are configured on the switch.

```
switch>show qos maps
   Number of Traffic Classes supported: 7
   Number of Transmit Queues supported: 7

   Cos-tc map:
     cos:  0  1  2  3  4  5  6  7
     ----------------------------
     tc:   1  0  2  3  4  4  5  6

   Dscp-tc map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ---------------------------------------
      0 :     0  0  0  0  0  0  0  0  1  1
      1 :     1  1  1  1  1  1  2  2  2  2
      2 :     2  2  2  2  3  3  3  3  3  3
      3 :     3  3  4  4  4  4  4  4  4  4
      4 :     4  4  4  4  4  4  4  4  5  5
      5 :     5  5  5  5  5  5  5  5  5  5
      6 :     5  5  5  5

   Tc-cos map:
     tc:   0  1  2  3  4  5  6
     -------------------------
     cos:  1  0  2  3  4  6  7

   Tc-queue map:
     tc:        0  1  2  3  4  5  6
     -----------------------------
     tx-queue:  0  1  2  3  4  5  6

switch>
```

tx-queue

The **tx-queue** command places the switch in Tx-queue configuration mode to configure a transmit queue on the configuration mode interface. Tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

FM4000 platform switches have eight queues, ***0*** through ***7***. Queue 7 is not exposed through the CLI and is not user configurable. Queue 7 is always mapped to traffic class 7, which is reserved for control traffic.

Petra platform switches have eight queues, ***0*** through ***7***, and all queues are exposed through the CLI. However, queue 7 is not user-configurable. Queue 7 is always mapped to traffic class 7, which is reserved for control traffic.

The **exit** command returns the switch to the originating interface configuration mode.

Command Mode

Interface-Ethernet configuration

Command Syntax

tx-queue *queue_level*

Parameters

- *queue_level* the transmit queue. Values range from ***0*** to ***6***.

Commands Available in tx-queue configuration mode

- **bandwidth percent (Transmit Queue Configuration Modes)**
- **priority (FM4000 and Petra platform switches)**
- **shape rate (Transmit Queue configuration modes)**

Restrictions

This command is available on FM4000 and Petra platform switches.

Guidelines

FM4000 platform switches: queues handle unicast and multicast traffic.

Petra platform switches: queues handle unicast traffic. Queues for multicast traffic are not supported.

Related Commands

- **mc-tx-queue**: Configures multicast transmit queues on trident platform switches.
- **uc-tx-queue**: Configures unicast transmit queues on trident platform switches.

Example

- This command enters Tx-queue configuration mode for transmit queue 3 of Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#tx-queue 3
switch(config-if-Et5-txq-3)#
```

uc-tx-queue

The **uc-tx-queue** command places the switch in uc-tx-queue configuration mode to configure a unicast transmit queue on the configuration mode interface. Uc-tx-queue configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

Trident switches have eight unicast queues (UC0 – UC7) and four multicast queues (MC0 – MC03), categorized into two priority groups. All queues are exposed through the CLI and are user-configurable.

- Priority Group 1: UC7, UC6, MC3
- Priority Group 0: UC5, UC4, MC2, UC3, UC2, MC1, UC1, UC0, MC0

The **exit** command returns the switch to the originating interface configuration mode.

Command Mode

Interface-Ethernet configuration

Command Syntax

uc-tx-queue *queue_level*

Parameters

- *queue_level* The multicast transmit queue number. Values range from ***0*** to ***7***.

Commands Available in tx-queue configuration mode:

- **bandwidth percent (Transmit Queue Configuration Modes)**
- **priority (Trident platform switches)**
- **shape rate (Transmit Queue configuration modes)**

Restrictions

This command is available on Trident platform switches.

Related Commands

- **mc-tx-queue**: Configures multiicast transmit queues on Trident platform switches.
- **tx-queue**: Configures transmit queues on FM4000 and Petra platform switches.

Example

- This command enters mc-tx-queue configuration mode for multicast transmit queue 3 of Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#uc-tx-queue 4
switch(config-if-Et5-mc-txq-4)#
```

Chapter 17

IPv4

Arista switches support Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) for routing packets across network boundaries. This chapter describes Arista's implementation of IPv4 and includes these sections:

- Section 17.1: Introduction
- Section 17.2: IPv4 Description
- Section 17.3: Configuring IPv4
- Section 17.4: IPv4 Command Descriptions

17.1 Introduction

Routing transmits network layer data packets over connected independent subnets. Each subnet is assigned an IP address range and each device on the subnet is assigned an IP address from that range. The connected subnets have IP address ranges that do not overlap.

A router is a network device that connects multiple subnets. Routers forward inbound packets to the subnet whose address range includes the packets' destination address. IPv4 and IPv6 are internet layer protocols that define packet-switched internetworking, including source-to-destination datagram transmission across multiple networks.

17.2 IPv4 Description

Internet Protocol version 4 (IPv4) is a communications protocol used for relaying network packets across a set of connected networks using the Internet Protocol suite. Each network device is assigned a 32-bit IP address that identifies its network location.

17.2.1 IPv4 Address Format

IPv4 addresses are composed of 32 bits, expressed in dotted decimal notation by four decimal numbers, each ranging from 0 to 255. A subnet is identified by an IP address and an address space defined by a routing prefix. The switch supports the following subnet formats:

- **IP address and subnet mask**: The subnet mask is a 32-bit number (dotted decimal notation) that specifies the subnet address space. The subnet address space is calculated by performing an AND operation between the IP address and subnet mask.

- **IP address and wildcard mask**: The wildcard mask is a 32-bit number (dotted decimal notation) that specifies the subnet address space. Wildcard masks differ from subnet masks in that the bits are inverted. Some commands use wildcard masks instead of subnet masks.
- **CIDR notation**: CIDR notation specifies the scope of the subnet space by using a decimal number to identify the number of leading ones in the routing prefix. When referring to wildcard notation, CIDR notation specifies the number of leading zeros in the routing prefix.

  Examples

  - These subnets (subnet mask and CIDR notation) are calculated identically:

    ```
    10.24.154.13 255.255.255.0
    10.24.154.13/24
    ```

    The defined space includes all addresses between ***10.24.154.0*** and ***10.24.154.255***.

  - These subnets (wildcard mask and CIDR notation) are calculated identically:

    ```
    124.17.3.142 0.0.0.15
    124.17.3.142/28
    ```

    The defined space includes all addresses between ***124.17.3.128*** and ***124.17.3.143***.

Section 17.3.2.1 describes the process of assigning an IP address to an interface.

17.2.2 Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP) is a protocol that maps IP addresses to MAC addresses that local network devices recognize. The ARP cache is a table that stores the correlated addresses of the devices for which the router facilitates data transmissions.

After receiving a packet, routers use ARP to find the MAC address of the device assigned to the packet's destination IP address. If the ARP cache contains both addresses, the router sends the packet to the specified port. If the ARP cache does not contain the addresses, ARP broadcasts a request packet to all devices in the subnet. The device at the requested IP address responds and provides its MAC address. ARP updates the ARP cache with a dynamic entry and forwards the packet to the responding device. Static ARP entries can also be added to the cache through the CLI.

Proxy ARP is an ARP variant. A network device (proxy) responds to ARP requests for network addresses on a different network with its MAC address. Traffic to the destination is directed to the proxy device which then routes the traffic toward the ultimate destination.

Section 17.3.4 describes the ARP configuration procedures.

17.2.3 DHCP Relay

DHCP relay is an agent that transmits Dynamic Host Configuration Protocol (DHCP) messages between clients and servers on different IP networks. A DHCP relay agent forwards DHCP requests that arrive on a configured interface to DHCP servers on separate subnets. Upon receiving a request, an agent generates and sends a DHCP packet. The new packet is based on the requesting packet, with the gateway address set and relay agent information inserted, as specified by the switch.

The switch can also be programmed to relay client DHCP packets forwarded from other relay agents without adding or modifying any relay agent options.

Section 17.3.5 describes the process of configuring DHCP relay.

17.3 Configuring IPv4

These sections describe IPv4 configuration tasks:

- Section 17.3.1: Enabling IPv4
- Section 17.3.2: IPv4 Routes and Addressing
- Section 17.3.3: Viewing IPv4 Network Components
- Section 17.3.4: Configuring ARP
- Section 17.3.5: Configuring DHCP Relay

17.3.1 Enabling IPv4

When IPv4 routing is enabled, the switch attempts to deliver inbound packets to destination IPv4 addresses by forwarding them to interfaces or next hop addresses specified by the forwarding table.

The **ip routing** command enables IPv4 routing.

Example

- This command enables IP routing:

```
Switch(config)#ip routing
Switch(config)#
```

17.3.2 IPv4 Routes and Addressing

17.3.2.1 Assigning an IPv4 Address to an Interface

The **ip address** command specifies the IPv4 address of an interface and the mask for the subnet to which the interface is connected.

Example

- These commands configure an IPv4 address with subnet mask for VLAN 200:

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ip address 10.0.0.1/24
switch(config-if-Vl200)#
```

17.3.2.2 Creating Static and Default IPv4 Routes

Static routes are entered through the CLI and are typically used when dynamic protocols are unable to establish routes to a specified destination prefix.

The **ip route** command creates a static route. The destination is a network segment; the source is either an IP address or a routable interface port. When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence.

Static routes have a default administrative distance of 1. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with a distance value of 200 is overridden by OSPF intra-area routes with a default distance of 110.

A route tage is a 32-bit number that is attached to a route. Route maps use tags to filter routes. Static routes have a default tag value of 0.

Example

- This command creates a static route:

```
Switch(config)#ip route 172.17.252.0/24 vlan 500
Switch(config)#
```

The default route denotes the packet forwarding rule that takes effect when no other route is configured for a specified IPv4 address. All packets with destinations that are not established in the routing table are sent to the destination specified by the default route.

The IPv4 default route source is 0.0.0.0/0. The default route destination is referred to as the default gateway.

Example

- This command creates a default route and establishes 192.14.0.4 as the default gateway address:

```
Switch(config)#ip route 0.0.0.0/0 192.14.0.4
Switch(config)#
```

17.3.2.3 Configuring ECMP (IPv4)

Equal cost multi-path (ECMP) is a routing strategy where traffic is forwarded over multiple paths that have equal routing metric values. Multiple routes that are configured to the same destination with the same administrative distance comprise an Equal Cost Multi-Path (ECMP) route. The switch attempts to spread outbound traffic to all ECMP route paths equally.

All ECMP paths are assigned the same tag value; commands that change the tag value of a path also change the tag value of all paths in the ECMP route.

In a network topology using ECMP routing, hash polarization may result when all switches perform identical hash calculations. Hash polarization leads to uneven load distribution among the data paths. Hash polarization is avoided when switches use different hash seeds to perform hash calculations.

The **ip load-sharing** command provides the hash seed to an algorithm that the switch uses to distribute data streams among multiple equal-cost routes to a specified subnet.

Example

- This command sets the IPv4 load sharing hash seed to two:

```
Switch(config)#ip load-sharing fm4000 2
Switch(config)#
```

17.3.3 Viewing IPv4 Network Components

Displaying the FIB and Routing Table

The **show ip route** command displays routing table entries that are in the forwarding information base (FIB), including static routes, routes to directly connected networks, and dynamically learned routes. Multiple equal-cost paths to the same prefix are displayed contiguously as a block, with the destination prefix displayed only on the first line.

The **show running-config** command displays configured commands not in the FIB. The **show ip route summary** command displays the number of routes, categorized by source, in the routing table.

Example

- This command displays IP routes learned through BGP.

```
switch>show ip route bgp
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, A - Aggregate

 B E    170.44.48.0/23 [20/0] via 170.44.254.78
 B E    170.44.50.0/23 [20/0] via 170.44.254.78
 B E    170.44.52.0/23 [20/0] via 170.44.254.78
 B E    170.44.54.0/23 [20/0] via 170.44.254.78
 B E    170.44.254.112/30 [20/0] via 170.44.254.78
 B E    170.53.0.34/32 [1/0] via 170.44.254.78
 B I    170.53.0.35/32 [1/0] via 170.44.254.2
                             via 170.44.254.13
                             via 170.44.254.20
                             via 170.44.254.67
                             via 170.44.254.35
                             via 170.44.254.98

switch>
```

- This command displays a summary of routing table contents.

```
switch>show ip route summary
Route Source         Number Of Routes
-------------------------------------
connected                    15
static                        0
ospf                         74
  Intra-area: 32 Inter-area:33 External-1:0 External-2:9
  NSSA External-1:0 NSSA External-2:0
bgp                           7
  External: 6 Internal: 1
internal                     45
attached                     18
aggregate                     0
switch>
```

Displaying Gateways

A gateway is a router that provides access to another network. The gateway of last resort, also known as the default route, is the route that a packet uses when the route to its destination address is unknown. The IPv4 default route in is 0.0.0.0/0.

The **show ip route gateway** command displays IP addresses of all gateways (next hops) used by active routes.

Example

- This command displays next hops used by active routes.

```
Switch>show ip route gateway
The following gateways are in use:
   172.25.0.1 Vlan101
   172.17.253.2 Vlan2000
   172.17.254.2 Vlan2201
   172.17.254.11 Vlan2302
   172.17.254.13 Vlan2302
   172.17.254.17 Vlan2303
   172.17.254.20 Vlan2303
   172.17.254.66 Vlan2418
   172.17.254.67 Vlan2418
   172.17.254.68 Vlan2768
   172.17.254.29 Vlan3020
Switch>
```

Displaying Host Routes

The **show ip route host** command displays all host routes in the host forwarding table. Host routes are those whose destination prefix is the entire address (mask = 255.255.255.255 or prefix = /32). Each displayed host route is labeled with its purpose:

- F static routes from the FIB.
- R routes defined because the IP address is an interface address.
- B broadcast address.
- A routes to any neighboring host for which the switch has an ARP entry.

Example

- This command displays all host routes in the host forwarding table.

```
Switch#show ip route host
R - receive B - broadcast F - FIB, A - attached

F   127.0.0.1 to cpu
B   172.17.252.0 to cpu
A   172.17.253.2 on Vlan2000
R   172.17.253.3 to cpu
A   172.17.253.10 on Vlan2000
R   172.17.254.1 to cpu
A   172.17.254.2 on Vlan2901
B   172.17.254.3 to cpu
B   172.17.254.8 to cpu
A   172.17.254.11 on Vlan2902
R   172.17.254.12 to cpu

F   172.26.0.28 via 172.17.254.20 on Vlan3003
                via 172.17.254.67 on Vlan3008
                via 172.17.254.98 on Vlan3492
via 172.17.254.86 on Vlan3884
                via 172.17.253.2 on Vlan3000
F   172.26.0.29 via 172.25.0.1 on Vlan101
F   172.26.0.30 via 172.17.254.29 on Vlan3910
F   172.26.0.31 via 172.17.254.33 on Vlan3911
F   172.26.0.32 via 172.17.254.105 on Vlan3912
Switch#
```

17.3.4 Configuring ARP

The switch uses ARP cache entries to correlate 32-bit IP addresses to 48-bit hardware addresses.

The **arp timeout** command specifies the duration of dynamic address entries in the Address Resolution Protocol (ARP) cache for addresses learned through the configuration mode interface. The default duration is 14400 seconds (four hours).

ARP entries added manually to the table do not age and must be manually removed.

Example

- This command specifies an ARP cache duration of 7200 seconds (two hours) for dynamic addresses added to the ARP cache that were learned through VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#arp timeout 7200
switch(config-if-Vl200)#show active
interface Vlan200
   arp timeout 7200
switch(config-if-Vl200)#
```

The **arp** command adds a static entry to an Address Resolution Protocol (ARP) cache.

Example

- This command adds a static entry to the ARP cache in the main VRF.

```
switch(config)#arp 172.22.30.52 0025.900e.c63c arpa
switch(config)#
```

The **show ip arp** command displays ARP cache entries that map an IP address to a corresponding MAC address. The table displays addresses by their host names when the command includes the ***resolve*** argument.

Example

- This command displays ARP cache entries that map MAC addresses to IP addresses.

```
switch>show ip arp
Address         Age (min)  Hardware Addr   Interface
172.25.0.2              0  004c.6211.021e  Vlan101, Port-Channel2
172.22.0.1              0  004c.6214.3699  Vlan1000, Port-Channel1
172.22.0.2              0  004c.6219.a0f3  Vlan1000, Port-Channel1
172.22.0.3              0  0045.4942.a32c  Vlan1000, Ethernet33
172.22.0.5              0  f012.3118.c09d  Vlan1000, Port-Channel1
172.22.0.6              0  00e1.d11a.a1eb  Vlan1000, Ethernet5
172.22.0.7              0  004f.e320.cd23  Vlan1000, Ethernet6
172.22.0.8              0  0032.48da.f9d9  Vlan1000, Ethernet37
172.22.0.9              0  0018.910a.1fc5  Vlan1000, Ethernet29
172.22.0.11             0  0056.cbe9.8510  Vlan1000, Ethernet26
switch>
```

17.3.5 Configuring DHCP Relay

Configuring the DHCP Relay Information Option (Global)

The **ip dhcp relay information option (Global)** command configures the switch to attach tags to DHCP requests before forwarding them to the DHCP servers designated by **ip helper-address** commands. The **ip dhcp relay information option circuit-id** command specifies the tag contents for packets forwarded by the interface that it configures.

Example

- This command enables the attachment of tags to DHCP requests that are forwarded to DHCP server addresses.

```
switch(config)#ip dhcp relay information option
switch(config)#
```

Configuring an Interface as a Relay Agent

The **ip helper-address** command enables the configuration mode interface as a DHCP relay agent and specifies a forwarding address for DHCP requests. An interface that is configured with multiple helper addresses forwards DHCP requests to all specified addresses.

The **ip dhcp relay information option circuit-id** command specifies the content of tags that the switch attaches to DHCP requests before they are forwarded from the configuration mode interface to DHCP server addresses specified by **ip helper-address** commands. Tags are attached to outbound DHCP requests only if the information option is enabled on the switch. The default value for each interface is the name and number of the interface.

Example

- These commands enable the DHCP relay agent on VLAN interface 200, configure the switch to forward DHCP requests received on this interface to the server at 10.10.41.15, and add x-1234 as the DHCP relay information option circuit-ID string.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ip helper-address 10.10.41.15
switch(config-if-Vl200)#ip dhcp relay information option circuit-id x-1234
switch(config-if-Vl200)#show active
interface Vlan200
   ip helper-address 10.10.41.15
   ip dhcp relay information option circuit-id x-1234
switch(config)(config-if-Vl200)#
```

DHCP Relay Switch Commands

The **ip dhcp relay always-on** command enables the switch DHCP relay agent on the switch regardless of the DHCP relay agent status on any interface. This programs the switch to relay all client DHCP packets forwarded from other relay agents.

Example

- This command enables the DHCP relay agent.

```
switch(config)#ip dhcp relay always-on
switch(config)#
```

17.4 IPv4 Command Descriptions

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Commands

- arp . . . . . Page 690
- clear ip dhcp relay counters . . . . . Page 694
- ip dhcp relay always-on . . . . . Page 698
- ip dhcp relay information option (Global) . . . . . Page 701
- ip dhcp relay log verbose . . . . . Page 703
- ip dhcp relay server . . . . . Page 704
- ip load-sharing . . . . . Page 706
- ip route . . . . . Page 709
- ip routing . . . . . Page 711

Interface Configuration Commands – Ethernet and Management Interfaces

- arp timeout . . . . . Page 691
- ip address . . . . . Page 696
- ip dhcp relay . . . . . Page 697
- ip dhcp relay circuit-id . . . . . Page 699
- ip dhcp relay client server . . . . . Page 700
- ip dhcp relay information option circuit-id . . . . . Page 702
- ip helper-address . . . . . Page 705
- ip local-proxy-arp . . . . . Page 707
- ip proxy-arp . . . . . Page 708

Privileged EXEC Commands

- clear arp-cache . . . . . Page 692
- clear ip arp . . . . . Page 693

EXEC Commands

- show arp . . . . . Page 712
- show ip . . . . . Page 713
- show ip arp . . . . . Page 714
- show ip dhcp relay . . . . . Page 716
- show ip dhcp relay counters . . . . . Page 717
- show ip helper-address . . . . . Page 718
- show ip interface . . . . . Page 719
- show ip interface brief . . . . . Page 720
- show ip route . . . . . Page 721
- show ip route gateway . . . . . Page 723
- show ip route host . . . . . Page 724
- show ip route summary . . . . . Page 725
- show ip route tag . . . . . Page 726

arp

The **arp** command adds a static entry to an Address Resolution Protocol (ARP) cache. The switch uses ARP cache entries to correlate 32-bit IP addresses to 48-bit hardware addresses.

The **no arp** and **default arp** commands remove the ARP cache entry with the specified IP address. When multiple VRFs contain ARP cache entries for identical IP addresses, each entry can only be removed individually.

Command Mode

Global Configuration

Command Syntax

```
arp [VRF_INSTANCE] ipv4_addr mac_addr arpa
no arp [VRF_INSTANCE] ipv4_addr
default arp [VRF_INSTANCE] ipv4_addr
```

Parameters

- *VRF_INSTANCE* specifies the VRF instance being modified.
  - <no parameter> changes are made to the main VRF.
  - **vrf** *vrf_name* changes are made to the specified user-defined VRF.
- *ipv4_addr* IPv4 address of ARP entry.
- *mac_addr* local data-link (hardware) address (48-bit dotted hex notation – H.H.H).

Examples

- This command adds a static entry to the ARP cache in the main VRF.

```
switch(config)#arp 172.22.30.52 0025.900e.c63c arpa
switch(config)#
```

- This command adds the same static entry to the ARP cache in the VRF named “purple.”

```
switch(config)#arp vrf purple 172.22.30.52 0025.900e.c63c arpa
switch(config)#
```

arp timeout

The **arp timeout** command specifies the duration of dynamic address entries in the Address Resolution Protocol (ARP) cache for addresses learned through the configuration mode interface. The default duration is 14400 seconds (four hours).

The **arp timeout** and **default arp timeout** commands restores the default ARP timeout for addresses learned on the configuration mode interface by deleting the corresponding **arp timeout** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
arp timeout arp_time
no arp timeout
default arp timeout
```

Parameters

- *arp_time* ARP timeout period (seconds). Values range from 60 to 65535. Default value is 14400.

Examples

- This command specifies an ARP cache duration of 7200 seconds (two hours) for dynamic addresses added to the ARP cache that were learned through VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#arp timeout 7200
switch(config-if-Vl200)#show active
interface Vlan200
   arp timeout 7200
switch(config-if-Vl200)#
```

clear arp-cache

The **clear arp-cache** command refreshes dynamic entries in the Address Resolution Protocol (ARP) cache. Refreshing the ARP cache updates IP address and MAC address mapping information in the ARP table and removes expired ARP entries not yet deleted by an internal, timer-driven process.

The command, without arguments, refreshes ARP cache entries for all enabled interfaces. With arguments, the command refreshes cache entries for the specified interface.

Command Mode

Privileged EXEC

Command Syntax

```
clear arp-cache [VRF_INSTANCE][INTERFACE_NAME]
```

Parameters

- *VRF_INSTANCE* specifies the VRF instance for which arp data is refreshed.
  - <no parameter> specifies the context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **main**.
- *INTERFACE_NAME* interface upon which ARP cache entries are refreshed. Options include:
  - <no parameter> All ARP cache entries.
  - **interface ethernet** *e_num* ARP cache entries of specified Ethernet interface.
  - **interface loopback** *l_num* ARP cache entries of specified loopback interface.
  - **interface management** *m_num* ARP cache entries of specified management interface.
  - **interface port-channel** *p_num* ARP cache entries of specified port-channel Interface.
  - **interface vlan** *v_num* ARP cache entries of specif VLAN interface.

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Example

- These commands display the ARP cache before and after ARP cache entries are refreshed.

```
switch#show arp
Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
172.22.30.118           0  001c.7301.6015  Management1
switch#clear arp-cache
switch#show arp
Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
switch#
```

clear ip arp

The **clear ip arp** command removes the specified dynamic ARP entry for the specified IP address from the Address Resolution Protocol (ARP) table.

Command Mode

Privileged EXEC

Command Syntax

clear ip arp [***VRF_INSTANCE***] *ipv4_addr*

Parameters

- *VRF_INSTANCE* specifies the VRF instance for which arp data is removed.
  - <no parameter> specifies the context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **main**.
- *ipv4_addr* IPv4 address of dynamic ARP entry.

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Example

- These commands display the ARP table before and after the removal of dynamic ARP entry for IP address 172.22.30.52.

```
switch#show arp
Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
172.22.30.52            0  0025.900e.c468  Management1
172.22.30.53            0  0025.900e.c63c  Management1
172.22.30.133           0  001c.7304.3906  Management1
Switch#clear ip arp 172.22.30.52
Switch#show arp
Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
172.22.30.53            0  0025.900e.c63c  Management1
172.22.30.133           0  001c.7304.3906  Management1
switch#
```

clear ip dhcp relay counters

The **clear ip dhcp relay counters** command resets the DHCP relay counters. The configuration mode determines which counters are reset:

- Global configuration: command clears the counters for the switch and for all interfaces.
- Interface configuration: command clears the counter for the configuration mode interface.

Command Mode

Global Configuration
Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
clear ip dhcp relay counters
```

Examples

- These commands clear the DHCP relay counters for VLAN 1045 and shows the counters before and after the **clear** command.

```
switch(config)#show ip dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  376  376    0 | 4 days, 19:55:12 ago
 All Resp |  277  277    0 |
          |                |
 Vlan1001 |  207  148    0 | 4 days, 19:54:24 ago
 Vlan1045 |  376  277    0 | 4 days, 19:54:24 ago

switch(config)#interface vlan 1045
switch(config-if-Vl1045)#clear ip dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  380  380    0 | 4 days, 21:19:17 ago
 All Resp |  281  281    0 |
          |                |
 Vlan1000 |  207  148    0 | 4 days, 21:18:30 ago
 Vlan1045 |    0    0    0 |          0:00:07 ago
```

- These commands clear all DHCP relay counters on the switch.

```
switch(config-if-Vl1045)#exit
switch(config)#clear ip dhcp relay counters
switch(config)#show ip dhcp relay counters

          |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop | Last Cleared
----------|----- ---- -----|-------------
  All Req |    0    0    0 |  0:00:03 ago
 All Resp |    0    0    0 |
          |                |
 Vlan1000 |    0    0    0 |  0:00:03 ago
 Vlan1045 |    0    0    0 |  0:00:03 ago

switch(config)#
```

ip address

The **ip address** command configures the IPv4 address and connected subnet on the configuration mode interface. Each interface can have one primary address and multiple secondary addresses.

The **no ip address** and **default ip address** commands remove the IPv4 address assignment from the configuration mode interface. Entering the command without specifying an address removes the primary and all secondary addresses from the interface. The primary address cannot be deleted until all secondary addresses are removed from the interface.

Removing all IPv4 address assignments from an interface disables IPv4 processing on that port.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip address ipv4_subnet [PRIORITY]
no ip address [ipv4_subnet] [PRIORITY]
default ip address [ipv4_subnet] [PRIORITY]
```

Parameters

- *ipv4_subnet* IPv4 and subnet address (CIDR or address-mask notation). ***Running-config*** stores value in CIDR notation.
- ***PRIORITY*** interface priority. Options include:
  - <no parameter> the address is the primary IPv4 address for the interface.
  - **secondary** the address is the secondary IPv4 address for the interface.

Guidelines

The **ip address** command is supported on routable interfaces.

Example

- This command configures an IPv4 address for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ip address 10.0.0.1/24
switch(config-if-Vl200)#
```

ip dhcp relay

The **ip dhcp relay** command is a component of the DHCP forwarder feature. This feature is superceded by the DHCP relay agent feature. DHCP forward commands are incompatible with the current implementation of DHCP relay agent.

The **no ip dhcp relay** and **default ip dhcp relay** commands remove the corresponding **ip dhcp relay** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip dhcp relay
no ip dhcp relay
default ip dhcp relay
```

Related Commands

The following commands support the current DHCP relay agent implementation.

- **ip helper-address**
- **ip dhcp relay always-on**
- **ip dhcp relay information option (Global)**
- **ip dhcp relay information option circuit-id**

The following commands are components of the defunct DHCP forwarder feature and must be removed from ***running-config*** before configuring the DHCP relay agent.

- **ip dhcp relay circuit-id**
- **ip dhcp relay client server**
- **ip dhcp relay log verbose**
- **ip dhcp relay server**

ip dhcp relay always-on

The **ip dhcp relay always-on** command enables the switch DHCP relay agent on the switch regardless of the DHCP relay agent status on any interface. By default, the DHCP relay agent is enabled only if at least one routable interface is configured with an **ip helper-address** statement.

The **no ip dhcp relay always-on** and **default ip dhcp relay always-on** commands remove the **ip dhcp relay always-on** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
ip dhcp relay always-on
no ip dhcp relay always-on
default ip dhcp relay always-on
```

Guidelines

The DHCP relay agent feature, of which the **ip dhcp relay always-on** command is a component, supercedes the defunct DHCP forward feature. The following commands are components of the DHCP forwarder feature and should be removed from *running-config* before configuring the DHCP relay agent.

- **ip dhcp relay**
- **ip dhcp relay circuit-id**
- **ip dhcp relay client server**
- **ip dhcp relay log verbose**
- **ip dhcp relay server**

Example

- This command enables the DHCP relay agent.

```
switch(config)#ip dhcp relay always-on
switch(config)#
```

ip dhcp relay circuit-id

The **ip dhcp relay circuit-id** command is a component of the DHCP forwarder feature. This feature is superceded by the DHCP relay agent feature. DHCP forward commands are incompatible with DHCP relay agent commands.

The **no ip dhcp relay circuit-id** and **default ip dhcp relay circuit-id** commands remove the corresponding **ip dhcp relay circuit-id** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip dhcp relay circuit-id id_label
no ip dhcp relay circuit-id
default ip dhcp relay circuit-id
```

Parameters

- ***id_label*** Circuit ID label. Format is alphanumeric characters (maximum 15 characters).

Related Commands

The following commands support the current DHCP relay agent implementation.

- **ip helper-address**
- **ip dhcp relay always-on**
- **ip dhcp relay information option (Global)**
- **ip dhcp relay information option circuit-id**

The following commands are components of the defunct DHCP forwarder feature and must be removed from ***running-config*** before configuring the DHCP relay agent.

- **ip dhcp relay**
- **ip dhcp relay server**
- **ip dhcp relay client server**
- **ip dhcp relay log verbose**

ip dhcp relay client server

The **ip dhcp relay client server** command is a component of the DHCP forwarder feature. This feature is superceded by the DHCP relay agent feature. DHCP forward commands are incompatible with DHCP relay agent commands.

The **no ip dhcp relay client server** and **default ip dhcp relay client server** commands remove the corresponding **ip dhcp relay client server** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip dhcp relay TYPE
no ip dhcp relay TYPE
default ip dhcp relay TYPE
```

Parameters

- *TYPE* Options include:
  - **client**
  - **server**
  - **client server**

Related Commands

The following commands support the current DHCP relay agent implementation.

- **ip helper-address**
- **ip dhcp relay always-on**
- **ip dhcp relay information option (Global)**
- **ip dhcp relay information option circuit-id**

The following commands are components of the defunct DHCP forwarder feature and must be removed from ***running-config*** before configuring the DHCP relay agent.

- **ip dhcp relay**
- **ip dhcp relay circuit-id**
- **ip dhcp relay server**
- **ip dhcp relay log verbose**

ip dhcp relay information option (Global)

The **ip dhcp relay information option** command configures the switch to attach tags to DHCP requests before forwarding them to the DHCP servers designated by **ip helper-address** commands. The **ip dhcp relay information option circuit-id** command specifies the tag contents for packets forwarded by the interface that it configures.

The **no ip dhcp relay information option** and **default ip dhcp relay information option** commands restore the switch's default setting of not attaching tags to DHCP requests by removing the **ip dhcp relay information option** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip dhcp relay information option
no ip dhcp relay information option
default ip dhcp relay information option
```

Guidelines

The DHCP relay agent feature, of which the **ip dhcp relay information option** command is a component, supercedes the defunct DHCP forward feature. The following commands are components of the DHCP forwarder feature and should be removed from ***running-config*** before configuring the DHCP relay agent.

- **ip dhcp relay**
- **ip dhcp relay circuit-id**
- **ip dhcp relay client server**
- **ip dhcp relay log verbose**
- **ip dhcp relay server**

Example

- This command enables the attachment of tags to DHCP requests that are forwarded to DHCP server addresses.

```
switch(config)#ip dhcp relay information option
switch(config)#
```

ip dhcp relay information option circuit-id

The **ip dhcp relay information option circuit-id** command specifies the content of tags that the switch attaches to DHCP requests before they are forwarded from the configuration mode interface to DHCP server addresses specified by **ip helper-address** commands. Tags are attached to outbound DHCP requests only if the information option is enabled on the switch (**ip dhcp relay information option circuit-id**. The default value for each interface is the name and number of the interface.

The **no ip dhcp relay information option circuit-id** and **default ip dhcp relay information option circuit-id** commands restore the default content setting for the configuration mode interface by removing the corresponding command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip dhcp relay information option circuit-id id_label
no ip dhcp relay information option circuit-id
default ip dhcp relay information option circuit-id
```

Parameters

- *id_label* Tag content. Format is alphanumeric characters (maximum 15 characters).

Guidelines

The DHCP relay agent feature, of which the **ip dhcp relay information option circuit-id** command is a component, supercedes the defunct DHCP forward feature. The following commands are components of the DHCP forwarder feature and should be removed from ***running-config*** before configuring the DHCP relay agent.

- **ip dhcp relay**
- **ip dhcp relay circuit-id**
- **ip dhcp relay client server**
- **ip dhcp relay log verbose**
- **ip dhcp relay server**

Example

- This command configures ***x-1234*** as the tag content for packets send from VLAN 200.

```
switch(config)#interface vlan 200
switch(config)(config-if-Vl200)#ip dhcp relay information option circuit-id
x-1234
switch(config)(config-if-Vl200)#
```

ip dhcp relay log verbose

The **ip dhcp relay log verbose** command is a component of the DHCP forwarder feature. This feature is superceded by the DHCP relay agent feature. DHCP forward commands are incompatible with DHCP relay agent commands.

The **no ip dhcp relay log verbose** and **default ip dhcp relay log verbose** commands remove the **ip dhcp relay log verbose** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip dhcp relay log verbose
no ip dhcp relay log verbose
default dhcp relay log verbose
```

Related Commands

The following commands support the current DHCP relay agent implementation.

- **ip helper-address**
- **ip dhcp relay always-on**
- **ip dhcp relay information option (Global)**
- **ip dhcp relay information option circuit-id**

The following commands are components of the defunct DHCP forwarder feature and should be removed from ***running-config*** before configuring the DHCP relay agent.

- **ip dhcp relay**
- **ip dhcp relay circuit-id**
- **ip dhcp relay client server**
- **ip dhcp relay server**

ip dhcp relay server

The **ip dhcp relay information relay server** command is a component of the DHCP forward feature. This feature is superceded by the DHCP relay agent feature. DHCP forward commands are incompatible with DHCP relay agent commands.

The **no ip dhcp relay information relay server** and **default dhcp relay information relay server** commands remove the corresponding **ip dhcp relay information relay server** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
ip dhcp relay server ipv4_addr
no ip dhcp relay server ipv4_addr
default ip dhcp relay server ipv4_addr
```

Parameters

- *ipv4_addr* IPv4 address.

Related Commands

The following commands support the current DHCP relay agent implementation.

- **ip helper-address**
- **ip dhcp relay always-on**
- **ip dhcp relay information option (Global)**
- **ip dhcp relay information option circuit-id**

The following commands are components of the defunct DHCP forwarder feature and should be removed from ***running-config*** before configuring the DHCP relay agent.

- **ip dhcp relay**
- **ip dhcp relay circuit-id**
- **ip dhcp relay client server**
- **ip dhcp relay log verbose**

ip helper-address

The **ip helper-address** command enables the DHCP relay agent on the configuration mode interface and specifies a forwarding address for DHCP requests. An interface that is configured with multiple helper-addresses forwards DHCP requests to all specified addresses.

The **no ip helper-address** and **default ip helper-address** commands remove the corresponding **ip helper-address** command from ***running-config***. Commands that do not specify an IP helper-address removes all helper-addresses from the interface.

Command Mode

Interface-Ethernet Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip helper-address ipv4_addr
no ip helper-address [ipv4_addr]
default ip helper-address [ipv4_addr]
```

Parameters

- *ipv4_addr* DHCP server address accessed by interface.

Guidelines

The DHCP relay agent feature, of which the **ip helper-address** command is a component, supercedes the defunct DHCP forward feature. The following commands are components of the DHCP forwarder feature and should be removed from ***running-config*** before configuring the DHCP relay agent.

- **ip dhcp relay**
- **ip dhcp relay circuit-id**
- **ip dhcp relay client server**
- **ip dhcp relay log verbose**
- **ip dhcp relay server**

Example

- This command enables the DHCP relay agent on VLAN interface 200 and configures the switch to forward DHCP requests received on this interface to the server at 10.10.41.15.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ip helper-address 10.10.41.15
switch(config-if-Vl200)#show active
interface Vlan200
   ip helper-address 10.10.41.15
switch(config-if-Vl200)#
```

ip load-sharing

The **ip load-sharing** command provides the hash seed to an algorithm that the switch uses to distribute data streams among multiple equal-cost routes to an individual IPv4 subnet.

In a network topology using Equal-Cost Multipath routing, all switches performing identical hash calculations may result in hash polarization, leading to uneven load distribution among the data paths. Hash polarization is avoided when switches use different hash seeds to perform different hash calculations.

The **no ip load-sharing** and **default ip load-sharing** commands return the hash seed to the default value of zero by removing the **ip load-sharing** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip load-sharing HARDWARE seed
no ip load-sharing HARDWARE
default ip load-sharing HARDWARE
```

Parameters

- *HARDWARE* The ASIC switching device. The available option depend on the switch platform. Verify available options with the CLI ? command.
  - **fm4000**
  - **petraA**
  - **trident**
- *seed* The hash seed. Value range varies by switch platform.

Example

- This command sets the IPv4 load sharing hash seed to one.

```
switch(config)#ip load-sharing fm4000 1
switch(config)#
```

ip local-proxy-arp

The **ip local-proxy-arp** command enables local proxy ARP (Address Resolution Protocol) on the configuration mode interface. Local proxy ARP programs the switch to respond to ARP requests for IP addresses within a subnet where routing is not normally required. A typical local proxy arp application is supporting isolated private VLANs that communicate with each other by routing packets.

The **no ip local-proxy-arp** and **default ip local-proxy-arp** commands disable local proxy ARP on the configuration mode interface by removing the corresponding **ip local-proxy-arp** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip local-proxy-arp
no ip local-proxy-arp
default ip local-proxy-arp
```

Examples

- These commands enable local proxy ARP on VLAN interface 140.

```
switch(config)#interface vlan 140
switch(config-if-Vl140)#ip local-proxy-arp
switch(config-if-Vl140)#show active
interface Vlan140
   ip local-proxy-arp
switch(config-if-Vl140)#
```

ip proxy-arp

The **ip proxy-arp** command enables proxy ARP on the configuration mode interface. Proxy ARP is disabled by default.

The **no ip proxy-arp** and **default ip proxy-arp** commands disable proxy ARP on the configuration mode interface by removing the corresponding **ip proxy-arp** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip proxy-arp
no ip proxy-arp
default ip proxy-arp
```

Examples

- This command enables proxy ARP on Ethernet interface 4.

```
switch(config)#interface ethernet 4
switch(config-if-Et4)#ip proxy-arp
switch(config-if-Et4)#
```

ip route

The **ip route** command creates a static route. The destination is a network segment; the source is either an IPv4 address or a routable port. When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence.

Static routes have a default administrative distance of 1. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with a distance value of 200 is overridden by OSPF intra-area routes with a default distance of 110.

Tags are used by route maps to filter routes. The default tag value on static routes is 0.

Multiple routes with the same destination and the same administrative distance comprise an Equal Cost Multi-Path (ECMP) route. The switch attempts to spread outbound traffic equally through all ECMP route paths. All paths comprising an ECMP are assigned identical tag values; commands that change the tag value of a path also change the tag value of all paths in the ECMP.

The **no ip route** and **default ip route** commands delete the specified static route by removing the corresponding **ip route** command from ***running-config***. Commands that do not list a source remove all **ip route** statements with the specified destination from ***running-config***. If **ip route** statements exist for the same IP address in multiple VRFs, each must be removed separately. All static routes in a user-defined VRF are deleted when the VRF is deleted.

Command Mode

Global Configuration

Command Syntax

```
ip route [VRF_INSTANCE] dest_net SOURCE [distance] [TAG_OPTION]
no ip route [VRF_INSTANCE] dest_net [SOURCE] [distance]
default ip route [VRF_INSTANCE] dest_net [SOURCE] [distance]
```

Parameters

- *dest_subnet* destination IPv4 subnet (CIDR or address-mask notation).
- ***VRF_INSTANCE*** specifies the VRF instance being modified.
  - <no parameter> changes are made to the main VRF.
  - **vrf** *vrf_name* changes are made to the specified VRF.
- ***SOURCE*** Route source. Options include:
  - *ipv4_addr* an IPv4 address.
  - **null0** Null0 interface.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
- *distance* administrative distance assigned to route. Options include:
  - <no parameter> route assigned default administrative distance of one.
  - <1-255> The administrative distance assigned to route.
- ***TAG_OPTION*** static route tag. Options include:
  - <no parameter> assigns default static route tag of 0.
  - **tag** *t_value* Static route tag value. *t_value* ranges from ***0*** to ***4294967295***.

Example

- This command creates a static route in the main VRF.

```
switch(config)#ip route 172.17.252.0/24 vlan 2000
switch(config)#
```

ip routing

The **ip routing** command enables IPv4 routing. When IPv4 routing is enabled, the switch attempts to deliver inbound packets to destination IPv4 addresses by forwarding them to interfaces or next hop addresses specified by the forwarding table.

The **no ip routing** and **default ip routing** commands disable IPv4 routing by removing the **ip routing** command from ***running-config***. When IPv4 routing is disabled, the switch attempts to deliver inbound packets to their destination MAC addresses. When this address matches the switch's MAC address, the packet is delivered to the CPU. IP packets with IPv4 destinations that differ from the switch's address are typically discarded. The **delete-static-routes** option removes static entries from the routing table.

IPv4 routing is disabled by default.

Command Mode

Global Configuration

Command Syntax

```
ip routing [VRF_INSTANCE]
no ip routing [DELETE_ROUTES] [VRF_INSTANCE]
default ip routing [DELETE_ROUTES] [VRF_INSTANCE]
```

Parameters

- ***DELETE_ROUTES*** Resolves routing table static entries when routing is disabled.
  - <no parameter> Routing table retains static entries.
  - **delete-static-routes** Static entries are removed from the routing table.
- ***VRF_INSTANCE*** specifies the VRF instance being modified.
  - <no parameter> changes are made to the main VRF.
  - **vrf** *vrf_name* changes are made to the specified user-defined VRF.

Example

- This command enables IPv4 routing.

```
switch(config)#ip routing
switch(config)#
```

show arp

The **show arp** command displays all ARP tables. This command differs from the **show ip arp** command in that it shows MAC bindings for all protocols, whereas **show ip arp** only displays MAC address – IP address bindings. Addresses are displayed as their host name by including the ***resolve*** argument.

Command Mode

EXEC

Command Syntax

show arp [***VRF_INSTANCE***] [***FORMAT***] [***HOST_ADDR***] [***HOST_NAME***] [***INT_NAME***] [***MAC_ADDR***]

Parameters

The ***VRF_INSTANCE*** and ***FORMAT*** parameters are always listed first and second, respectively. All other parameters can be placed in any order.

- ***VRF_INSTANCE*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **main**.
- ***FORMAT*** Display format of host address. Options include:
  - <no parameter> entries associate hardware address with an IPv4 address.
  - **resolve** entry associate hardware address with a host name (if it exists).
- ***HOST_ADDR*** IPv4 address by which routing table entries are filtered. Options include:
  - <no parameter> routing table entries are not filtered by host address.
  - *ipv4_addr* table entries matching specified IPv4 address.
- ***HOST_NAME*** Host name by which routing table entries are filtered. Options include:
  - <no parameter> routing table entries are not filtered by host name.
  - **host** *hostname* entries matching *hostname* (text).
- ***INTERFACE_NAME*** interfaces for which command displays status.
  - <no parameter> Routing table entries are not filtered by interface.
  - **interface ethernet** *e_num* Routed Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Routed loopback interface specified by *l_num*.
  - **interface management** *m_num* Routed management interface specified by *m_num*.
  - **interface port-channel** *p_num* Routed port channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.
- ***MAC_ADDR*** MAC address by which routing table entries are filtered. Options include:
  - <no parameter> Routing table entries are not filtered by interface MAC address.
  - **mac_address** *mac_address* entries matching *mac_address* (dotted hex notation – H.H.H).

Example

- This command displays the ARP table.

```
switch>show ip arp
Address         Age (min)  Hardware Addr   Interface
172.22.30.1             0  001c.730b.1d15  Management1
172.22.30.133           0  001c.7304.3906  Management1
switch>
```

show ip

The **show ip** command displays IPv4 routing, IPv4 multicast routing, and VRRP status on the switch.

Command Mode
EXEC

Command Syntax
show ip

Example

- This command displays IPv4 routing status.

```
switch>show ip

IP Routing : Enabled
IP Multicast Routing : Disabled
VRRP: Configured on 0 interfaces
switch>
```

show ip arp

The **show ip arp** command displays ARP cache entries that map an IPv4 address to a corresponding MAC address. The table displays addresses by their host names when the command includes the ***resolve*** argument.

Command Mode

EXEC

Command Syntax

```
show ip arp [VRF_INSTANCE] [FORMAT] [HOST_ADDR] [HOST_NAME] [INT_NAME] [MAC_ADDR]
```

Parameters

The ***VRF_INSTANCE*** and ***FORMAT*** parameters are always listed first and second, respectively. All other parameters can be placed in any order.

- ***VRF_INSTANCE*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **main**.
- ***FORMAT*** Display format of host address. Options include:
  - <no parameter> entries associate hardware address with an IPv4 address.
  - **resolve** entry associate hardware address with a host name (if it exists).
- ***HOST_ADDR*** IPv4 address by which routing table entries are filtered. Options include:
  - <no parameter> routing table entries are not filtered by host address.
  - *ipv4_addr* table entries matching specified IPv4 address.
- ***HOST_NAME*** Host name by which routing table entries are filtered. Options include:
  - <no parameter> routing table entries are not filtered by host name.
  - **host** *hostname* entries matching *hostname* (text).
- ***INTERFACE_NAME*** interfaces for which command displays status.
  - <no parameter> Routing table entries are not filtered by interface.
  - **interface ethernet** *e_num* Routed Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Routed loopback interface specified by *l_num*.
  - **interface management** *m_num* Routed management interface specified by *m_num*.
  - **interface port-channel** *p_num* Routed port channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.
- ***MAC_ADDR*** MAC address by which routing table entries are filtered. Options include:
  - <no parameter> Routing table entries are not filtered by interface MAC address.
  - **mac_address** *mac_address* entries matching *mac_address* (dotted hex notation – H.H.H).

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Examples

- This command displays ARP cache entries that map MAC addresses to IPv4 addresses.

```
switch>show ip arp
Address         Age (min)  Hardware Addr   Interface
172.25.0.2              0  004c.6211.021e  Vlan101, Port-Channel2
172.22.0.1              0  004c.6214.3699  Vlan1000, Port-Channel1
172.22.0.2              0  004c.6219.a0f3  Vlan1000, Port-Channel1
172.22.0.3              0  0045.4942.a32c  Vlan1000, Ethernet33
172.22.0.5              0  f012.3118.c09d  Vlan1000, Port-Channel1
172.22.0.6              0  00e1.d11a.a1eb  Vlan1000, Ethernet5
172.22.0.7              0  004f.e320.cd23  Vlan1000, Ethernet6
172.22.0.8              0  0032.48da.f9d9  Vlan1000, Ethernet37
172.22.0.9              0  0018.910a.1fc5  Vlan1000, Ethernet29
172.22.0.11             0  0056.cbe9.8510  Vlan1000, Ethernet26
switch>
```

- This command displays ARP cache entries that map a MAC addresses to IPv4 addresses. Host names assigned to IP addresses are displayed in place of the address.

```
switch>show ip arp resolve
Address         Age (min)  Hardware Addr   Interface
green-vl101.new         0  004c.6211.021e  Vlan101, Port-Channel2
172.22.0.1              0  004c.6214.3699  Vlan1000, Port-Channel1
orange-vl1000.n         0  004c.6219.a0f3  Vlan1000, Port-Channel1
172.22.0.3              0  0045.4942.a32c  Vlan1000, Ethernet33
purple.newcompa         0  f012.3118.c09d  Vlan1000, Port-Channel1
pink.newcompany         0  00e1.d11a.a1eb  Vlan1000, Ethernet5
yellow.newcompa         0  004f.e320.cd23  Vlan1000, Ethernet6
172.22.0.8              0  0032.48da.f9d9  Vlan1000, Ethernet37
royalblue.newco         0  0018.910a.1fc5  Vlan1000, Ethernet29
172.22.0.11             0  0056.cbe9.8510  Vlan1000, Ethernet26
switch>
```

show ip dhcp relay

The **show ip dhcp relay** command displays the DHCP relay agent configuration status on the switch.

Command Mode
EXEC

Command Syntax
show ip dhcp relay

Example

- This command displays the DHCP relay agent configuration status.

```
switch>show ip dhcp relay
DHCP servers: 172.22.22.11
Vlan1000:
  DHCP clients are permitted on this interface
Vlan1036:
  DHCP clients are permitted on this interface
DHCP server responses are processed on all interfaces
DHCP Relay log level: default
switch>
```

show ip dhcp relay counters

The **show ip dhcp relay counters** command displays the number of DHCP packets received, forwarded, or dropped on the switch and on all interfaces enabled as DHCP relay agents.

Command Mode
EXEC

Command Syntax
show ip dhcp relay

Example

- This command displays the IP DHCP relay counter table.

```
switch>show ip dhcp relay counters

           |  Dhcp Packets  |
Interface | Rcvd Fwdd Drop |         Last Cleared
----------|----- ---- -----|---------------------
  All Req |  376  376    0 | 4 days, 19:55:12 ago
 All Resp |  277  277    0 |
          |                |
 Vlan1000 |    0    0    0 | 4 days, 19:54:24 ago
 Vlan1036 |  376  277    0 | 4 days, 19:54:24 ago

switch>
```

show ip helper-address

The **show ip helper-address** command displays the status of DHCP relay agent parameters on the switch and each interface where at least one feature parameter is listed. The command provides status on the following parameters:

- Global: DHCP relay agent Always-on mode, DHCP relay agent Information option
- Interface: DHCP server (list of addresses), Circuit ID contents

Command Mode

EXEC

Command Syntax

```
show ip helper-address
```

Example

- This command displays the DHCP Agent Relay parameter status.

```
switch>show ip helper-address
DHCP Relay Agent Information Option Enabled
DHCP Relay Agent Always-On Mode Enabled
Interface: Vlan200
   Circuit ID: V-200
   DHCP servers: 10.3.31.14
switch>
```

show ip interface

The **show ip interface** command displays the status of specified interfaces that are configured as routed ports. The command provides the following information:

- Interface description
- Internet address
- Broadcast address
- Address configuration method
- Proxy-ARP status
- MTU size

Command Mode

EXEC

Command Syntax

show ip interface [***INTERFACE_NAME***]

Parameters

- *INTERFACE_NAME* interfaces for which command displays status.
  - — <no parameter> all routed interfaces
  - — **ethernet** *e_range* Routed Ethernet interfaces specified by *e_range*.
  - — **loopback** *l_range* Routed loopback interfaces specified by *l_range*.
  - — **management** *m_range* Routed management interfaces specified by *m_range*.
  - — **port-channel** *p_range* Routed port channel Interfaces specified by *p_range*.
  - — **vlan** *v_range* VLAN interfaces specified by *v_range*.

Example

- This command displays IP status of configured VLAN interfaces numbered between 900 and 910.

```
switch>show ip interface vlan 900-910
! Some interfaces do not exist
Vlan901 is up, line protocol is up (connected)
  Description: ar.pqt.mlag.peer
  Internet address is 170.23.254.1/30
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  Proxy-ARP is disabled
  MTU 9212 bytes
Vlan902 is up, line protocol is up (connected)
  Description: ar.pqt.b.170.23.254.8/29
  Internet address is 170.23.254.12/29
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  Proxy-ARP is disabled
  MTU 9212 bytes
Vlan903 is up, line protocol is up (connected)
  Description: ar.pqt.rn.170.23.254.16/29
  Internet address is 170.23.254.19/29
  Broadcast address is 255.255.255.255
  Address determined by manual configuration
  Proxy-ARP is disabled
  MTU 9212 bytes
```

show ip interface brief

The **show ip interface brief** command displays the status summary of the specified interfaces that are configured as routed ports. The command provides the following information for each specified interface:

- IP address
- Operational status
- Line protocol status
- MTU size

Command Mode

EXEC

Command Syntax

show ip interface [***INTERFACE_NAME***] **brief**

Parameters

- ***INTERFACE_NAME*** interfaces for which command displays status.
  - — <no parameter> all routed interfaces.
  - — **ethernet** *e_range* Routed Ethernet interfaces specified by *e_range*.
  - — **loopback** *l_range* Routed loopback interfaces specified by *l_range*.
  - — **management** *m_range* Routed management interfaces specified by *m_range*.
  - — **port-channel** *p_range* Routed port channel Interfaces specified by *p_range*.
  - — **vlan** *v_range* VLAN interfaces specified by *v_range*.

Example

- This command displays the summary status of VLAN interfaces 900-910

```
switch>show ip interface vlan 900-910 brief
! Some interfaces do not exist
Interface              IP Address         Status     Protocol         MTU
Vlan901                170.33.254.1/30    up         up              9212
Vlan902                170.33.254.14/29   up         up              9212
Vlan905                170.33.254.17/29   up         up              1500
Vlan907                170.33.254.67/29   up         up              9212
Vlan910                170.33.254.30/30   up         up              9212
```

show ip route

The **show ip route** command displays routing table entries that are in the forwarding information base (FIB), including static routes, routes to directly connected networks, and dynamically learned routes. Multiple equal-cost paths to the same prefix are displayed contiguously as a block, with the destination prefix displayed only on the first line.

The **show running-config** command displays configured commands not in the FIB.

Command Mode

EXEC

Command Syntax

```
show ip route [VRF_INSTANCE] [ADDRESS] [ROUTE_TYPE] [INFO_LEVEL] [PREFIX]
```

Parameters

The *VRF_INSTANCE* and *ADDRESS* parameters are always listed first and second, respectively. All other parameters can be placed in any order.

- *VRF_INSTANCE* specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **main**.
- *ADDRESS* Filters routes by IPv4 address or subnet.
  - <no parameter> all routing table entries.
  - *ipv4_addr* routing table entries matching specified address.
  - *ipv4_subnet* routing table entries matching specified subnet (CIDR or address-mask).
- *ROUTE_TYPE* Filters routes by specified protocol or origin. Options include:
  - <no parameter> all routing table entries.
  - **aggregate** entries for BGP aggregate routes.
  - **bgp** entries added through BGP protocol.
  - **connected** entries for routes to networks directly connected to the switch.
  - **kernel** entries appearing in Linux kernel but not added by EOS software.
  - **ospf** entries added through OSPF protocol.
  - **rip** entries added through RIP protocol.
  - **static** entries added through CLI commands.
- *INFO_LEVEL* Filters entries by next hop connection. Options include:
  - <no parameter> filters routes whose next hops are directly connected.
  - **detail** displays all routes.
- *PREFIX* filters routes by prefix.
  - <no parameter> specific route entry that matches the ADDRESS parameter.
  - **longer-prefixes** all subnet route entries in range specified by ADDRESS parameter.

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Example

- This command displays IPv4 routes learned through BGP.

```
switch>show ip route bgp
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, A - Aggregate

 B E    170.44.48.0/23 [20/0] via 170.44.254.78
 B E    170.44.50.0/23 [20/0] via 170.44.254.78
 B E    170.44.52.0/23 [20/0] via 170.44.254.78
 B E    170.44.54.0/23 [20/0] via 170.44.254.78
 B E    170.44.254.112/30 [20/0] via 170.44.254.78
 B E    170.53.0.34/32 [1/0] via 170.44.254.78
 B I    170.53.0.35/32 [1/0] via 170.44.254.2
                             via 170.44.254.13
                             via 170.44.254.20
                             via 170.44.254.67
                             via 170.44.254.35
                             via 170.44.254.98
```

show ip route gateway

The **show ip route gateway** command displays IP addresses of all gateways (next hops) used by active routes.

Command Mode
EXEC

Command Syntax
show ip route gateway [*VRF_INSTANCE*]

Parameters

- *VRF_INSTANCE* specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **main**.

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Example

- This command displays next hops used by active routes.

```
switch>show ip route gateway
The following gateways are in use:
   172.25.0.1 Vlan101
   172.17.253.2 Vlan3000
   172.17.254.2 Vlan3901
   172.17.254.11 Vlan3902
   172.17.254.13 Vlan3902
   172.17.254.17 Vlan3903
   172.17.254.20 Vlan3903
   172.17.254.66 Vlan3908
   172.17.254.67 Vlan3908
   172.17.254.68 Vlan3908
   172.17.254.29 Vlan3910
   172.17.254.33 Vlan3911
   172.17.254.35 Vlan3911
   172.17.254.105 Vlan3912
   172.17.254.86 Vlan3984
   172.17.254.98 Vlan3992
   172.17.254.99 Vlan3992
switch>
```

show ip route host

The **show ip route host** command displays all host routes in the host forwarding table. Host routes are those whose destination prefix is the entire address (mask = 255.255.255.255 or prefix = /32). Each entry includes a code of the route's purpose:

- F static routes from the FIB.
- R routes defined because the IP address is an interface address.
- B broadcast address.
- A routes to any neighboring host for which the switch has an ARP entry.

Command Mode

EXEC

Command Syntax

show ip route host [*VRF_INSTANCE*]

Parameters

- ***VRF_INSTANCE*** specifies the VRF instance for which data is displayed.
  - <no parameter> context-active VRF.
  - **vrf** *vrf_name* specifies name of VRF instance. System default VRF is specified by **main**.

Related Commands

- **routing-context vrf** specifies the context-active VRF.

Example

- This command displays all host routes in the host forwarding table.

```
switch>show ip route host
R - receive B - broadcast F - FIB, A - attached

F   127.0.0.1 to cpu
B   172.17.252.0 to cpu
A   172.17.253.2 on Vlan2000
R   172.17.253.3 to cpu
A   172.17.253.10 on Vlan2000
B   172.17.253.255 to cpu
B   172.17.254.0 to cpu
R   172.17.254.1 to cpu
B   172.17.254.3 to cpu
B   172.17.254.8 to cpu
A   172.17.254.11 on Vlan2902
R   172.17.254.12 to cpu

F   172.26.0.28 via 172.17.254.20 on Vlan3003
                via 172.17.254.67 on Vlan3008
                via 172.17.254.98 on Vlan3492
                via 172.17.254.2 on Vlan3601
                via 172.17.254.13 on Vlan3602
via 172.17.253.2 on Vlan3000
F   172.26.0.29 via 172.25.0.1 on Vlan101
F   172.26.0.30 via 172.17.254.29 on Vlan3910
F   172.26.0.32 via 172.17.254.105 on Vlan3912
switch>
```

show ip route summary

The **show ip route summary** command displays the number of routes, categorized by source, in the routing table.

Command Mode
EXEC

Command Syntax
show ip route summary

Example

- This command displays a summary of the routing table contents.

```
switch>show ip route summary
Route Source         Number Of Routes
-------------------------------------
connected                   15
static                       0
ospf                        74
  Intra-area: 32 Inter-area:33 External-1:0 External-2:9
  NSSA External-1:0 NSSA External-2:0
bgp                          7
  External: 6 Internal: 1
internal                    45
attached                    18
aggregate                    0
switch>
```

show ip route tag

The **show ip route tag** command displays the route tag assigned to the specified IPv4 address or subnet. Route tags are added to static routes for use by route maps.

Command Mode
EXEC

Command Syntax
show ip route *ADDRESS* tag

Parameters

- *ADDRESS* displays routes of specified IPv4 address or subnet.
  - *ipv4_addr* routing table entries matching specified IPv4 address.
  - *ipv4_subnet* routing table entries matching specified IPv4 subnet (CIDR or address-mask).

Example

- This command displays the route tag for the specified subnet.

```
switch>show ip route 172.17.50.0/23 tag
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, A - Aggregate

 O E2   172.17.50.0/23 tag 0

switch>
```

Chapter 18

IPv6

Arista switches support Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) for routing packets across network boundaries. This chapter describes Arista's implementation of IPv6 and includes these sections:

- Section 18.1: Introduction
- Section 18.2: IPv6 Description
- Section 18.3: Configuring IPv6
- Section 18.4: IPv6 Command Descriptions

18.1 Introduction

Routing transmits network layer data packets over connected independent subnets. Each subnet is assigned an IP address range and each device on the subnet is assigned an IP address from that range.

Connected subnets have IP address ranges that do not overlap. A router is a network device connecting multiple subnets. Routers forward inbound packets to the subnet whose address range includes the packets' destination address.

IPv4 and IPv6 are internet layer protocols that define packet-switched internetworking, including source-to-destination datagram transmission across multiple networks. The switch supports IP Version 4 (IPv4) and IP Version 6 (IPv6).

IPv6 is described by RFC 2460: Internet Protocol, Version 6 (IPv6) Specification. RFC 2463 describes the Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. ICMPv6 is a core protocol of the Internet Protocol suite.

18.2 IPv6 Description

Internet Protocol Version 6 is a communications protocol used for relaying network packets across a set of connected networks using the Internet Protocol suite. Each network device is assigned a 128 bit IP address that identifies its network location.

IPv6 specifies a packet format that minimizes router processing of packet headers. Since the IPv4 and IPv6 packet headers differ significantly, the protocols are not interoperable. Many transport and application-layer protocols require little or no change to operate over IPv6.

18.2.1 IPv6 Address Format

IPv6 addresses have 128 bits, represented by eight 16-bit hexadecimal numbers separated by colons. IPv6 addresses are abbreviated as follows:

- Leading zeroes in each 16-bit number may be omitted.
- One set of consecutive 16-bit numbers that equal zero may be replaced by a double colon.

Example

- The following three IPv6 hexadecimal number representations refer to the same address:

```
d28e:0000:0000:0000:0234:812f:61ed:4419
d28e:0:0:0:234:812f:61ed:4419
d28e::234:812f:61ed:4419
```

IPv6 addresses typically denote a 64-bit network prefix and a 64-bit host address. The format of an IPv6 address denotes its networking methodology.

Unicast and Anycast Addressing

Unicast addressing defines a one-to-one association between the destination address and a network endpoint. Each destination address uniquely identifies a single receiver endpoint. Anycast addressing defines a one-to-one-of-many association: packets to a single member of a group of potential receivers identified by the same destination address.

Unicast and anycast addresses are typically composed as follows:

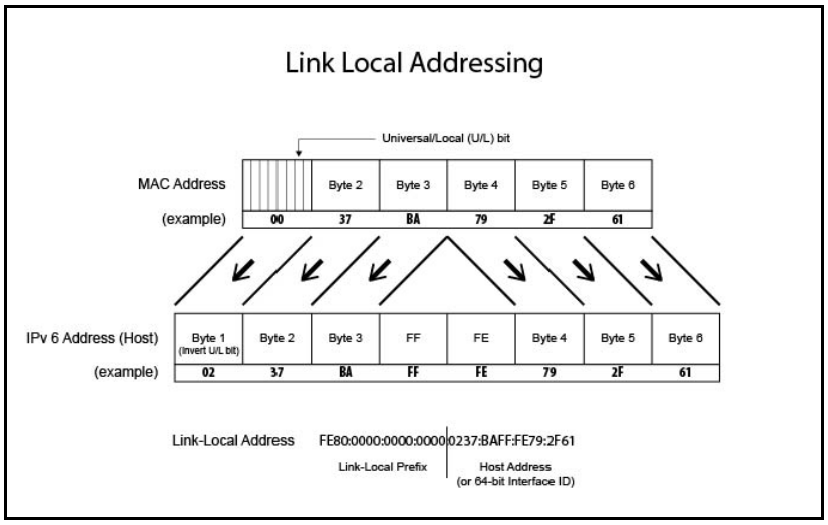
- a 64-bit network prefix that identifies the network segment
- a 64-bit interface identifier that is based on interface MAC address

The format of a network address identifies the scope of the address

- Global address: valid in all networks and connect with other addresses with global scope anywhere or to addresses with link-local scope on the directly attached network.
- Link-local address: scope extends only to the link to which the interface is directly connected. Link-local addresses are not routable off the link.

  Link-local addresses are created by the switch and are not configurable. Figure 18-1 depicts the switch's link local address derivation method.

Figure 18-1 Link Local Address Derivation



Multicast Addressing

Multicast addressing defines a one-to-many association: packets are simultaneously routed from a single sender to multiple endpoints in a single transmission. The network replicates packets as required by network links that contain a recipient endpoint. One multicast address is assigned to an interface for each multicast group to which the interface belongs.

A solicited-node multicast address is an IPv6 multicast address scope extends only to the link to which the interface is directly connected. All IPv6 hosts have at least one such address per interface. Solicited-node multicast addresses are used by the Neighbor Discovery Protocol to obtain layer 2 link-layer addresses of other nodes.

18.2.2 Neighbor Discovery Protocol

The Neighbor Discovery Protocol (RFC 4861) operates with IPv6 to facilitate the following tasks for nodes within a specified prefix space:

- autoconfiguring a node's IPv6 address
- sensing other nodes on the link
- discovering the link-local addresses of other nodes on the link
- detecting duplicate addresses
- discovering available routers
- discovering DNS servers
- discovering the link's address prefix
- maintaining path reachability data to other active neighbor nodes

The Neighbor Discovery Protocol protocol defines five different ICMPv6 packet types:

- Router Solicitation
- Router Advertisement
- Neighbor Solicitation
- Neighbor Advertisement
- Redirect

18.3 Configuring IPv6

These sections describe IPv6 configuration tasks:

- Section 18.3.1: Configuring IPv6 on the Switch
- Section 18.3.2: Configuring IPv6 on an Interface
- Section 18.3.3: Viewing IPv6 Network Components

18.3.1 Configuring IPv6 on the Switch

18.3.1.1 Enabling IPv6 Unicast Routing on the Switch

The **ipv6 unicast-routing** command enables the forwarding of IPv6 unicast packets. When routing is enabled, the switch attempts to deliver inbound packets to destination addresses by forwarding them to interfaces or next hop addresses specified by the IPv6 routing table.

Example

- This command enables IPv6 unicast-routing.

```
Switch(config)#ipv6 unicast-routing
Switch(config)#
```

18.3.1.2 Configuring Default and Static IPv6 Routes

The **ipv6 route** command creates an IPv6 static route. The destination is a IPv6 prefix; the source is an IPv6 address or a routable interface port. When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence.

Static routes have a default administrative distance of 1. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with a distance value of 200 is overridden by OPSF intra-area routes with a default distance of 110.

Multiple routes that are configured to the same destination with the same administrative distance comprise an Equal Cost Multi-Path (ECMP) route. The switch attempts to spread outbound traffic to all ECMP route paths equally. All ECMP paths are assigned the same tag value; commands that change the tag value of any ECMP path change the tag value of all paths in the ECMP.

Example

- This command creates an IPv6 static route.

```
Switch(config)#ipv6 route 10:23:31:00:01:32:93/24 vlan 300
Switch(config)#
```

The default route denotes the packet forwarding rule that takes effect when no other route is configured for a specified IPv6 address. All packets with destinations that are not established in the routing table are sent to the destination specified by the default route.

The IPv6 default route source is ::/0. The default route destination is referred to as the default gateway.

Example

- This command creates a default route and establishes fd7a:629f:52a4:fe61::2 as the default gateway address.

```
Switch(config)#ipv6 route ::/0 fd7a:629f:52a4:fe61::2
Switch(config)#
```

18.3.2 Configuring IPv6 on an Interface

18.3.2.1 Enabling IPv6 on an Interface

The **ipv6 enable** command enables IPv6 processing on the configuration mode interface if it not configured with an explicit IPv6 address. The command also configures an IPv6 link-local unicast address on the interface.

The **no ipv6 enable** command disables IPv6 processing on an interface that has not been configured with an explicit IPv6 address. The command does not disable IPv6 processing on an interface that is configured with an explicit IPv6 address.

Example

- This command enables IPv6 processing on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-vl200)#ipv6 enable
switch(config-vl200)#
```

18.3.2.2 Assigning an IPv6 Address to an Interface

The **ipv6 address** command enables IPv6 processing on the configuration mode interface, assigns a global IPv6 address to the interface, and defines the prefix length. This command is supported on routable interfaces. Multiple global IPv6 addresses can be assigned to an interface.

Example

- These commands configure an IPv6 address with subnet mask for VLAN 200:

```
Switch(config)#interface vlan 200
Switch(config-if-Vl200)#ipv6 address 10:23:31:00:01:32:93/24
Switch(config-if-Vl200)#
```

18.3.2.3 IPv6 Neighbor Discovery

The IPv6 Neighbor Discovery protocol defines a method for nodes to perform the following network maintenance tasks:

- determine layer 2 addresses for neighbors known to reside on attached links
- detect changed layer 2 addresses
- purge invalid values from the neighbor cache table
- (hosts) find neighboring routers to forward packets
- track neighbor reachability status

IPv6 Neighbor Discovery is defined by RFC 2461. IPv6 Stateless Address Autoconfiguration is described by RFC 2462.

The following sections describe Neighbor Discovery configuration tasks.

Reachable Time

The **ipv6 nd reachable-time** command specifies the time period that the switch includes in the reachable time field of Router Advertisements (RAs) sent from the configuration mode interface. The reachable time defines the period that a remote IPv6 node is considered reachable after a reachability confirmation event.

Example

- These commands configure the entry of 25000 (25 seconds) in the reachable time field of RAs sent from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd reachable-time 25000
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 address fd7a:4321::1/64
   ipv6 nd reachable-time 25000
switch(config-if-Vl200)#
```

Router Advertisement Interval

The **ipv6 nd ra interval** command configures the interval between IPv6 router advertisement (RA) transmissions from the configuration mode interface.

Example

- These commands configure a router advertisement transmission interval of 60 seconds on VLAN interface 200, then displays the interface status.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra interval 60
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 nd ra interval 60
switch(config-if-Vl200)#
```

Router Lifetime

The **ipv6 nd ra lifetime** command specifies the value that the switch places in the ***router lifetime*** field of IPv6 router advertisements (RAs) sent from the configuration mode interface. The router lifetime specifies the period that the router can be considered as a default router by RA recipients.

Setting the value to 0 indicates that the router should not be considered a default router on this interface. The ***router lifetime*** value can be set to a nonzero value to indicate that it should be considered a default router on this interface. The nonzero value for the ***router lifetime*** value should not be less than the router advertisement interval

Example

- This command configures the switch to enter 2700 in the router lifetime field of RAs tranmitted from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra lifetime 2700
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 nd ra lifetime 2700
switch(config-if-Vl200)#
```

Router Advertisement Prefix

The **ipv6 nd prefix** command configures neighbor discovery router advertisement prefix inclusion for router advertisements (RAs) sent from the configuration mode interface.

By default, all prefixes configured as IPv6 addresses are advertised in the interface's RAs. The **ipv6 nd prefix** command with the ***no-advertise*** option prevents advertising of the specified prefix without affecting the advertising of other prefixes specified as IPv6 addresses. When an interface configuration includes at least one **ipv6 nd prefix** command that enables prefix advertising, RAs advertise only prefixes specified through **ipv6 nd prefix** commands.

Commands enabling prefix advertising also specify the advertised valid and preferred lifetime periods. Default periods are 2,592,000 (valid) and 604,800 (preferred) seconds.

Example

- These commands enable neighbor discovery advertising for IPv6 address 3012:D678::/64, specifying a valid lifetime of 1,296,000 seconds and the default preferred lifetime.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd prefix 3012:D678::/64 1296000
switch(config-if-Vl200)#
```

Router Advertisement Suppression

The **ipv6 nd ra suppress** command suppress IPv6 router advertisement transmissions on the configuration mode interface. By default, only unsolicited RAs that are transmitted periodically are suppressed. The **all** option configures the switch to suppress all RAs, including those responding to a router solicitation.

Example

- This command suppresses all RAs on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-vl200)#ipv6 nd ra suppress all
switch(config-vl200)#
```

Router Advertisement MTU Suppression

The **ipv6 nd ra mtu suppress** command suppress the Router Advertisement MTU option on the configuration mode interface. The MTU option ensures that all nodes on a link use the same MTU value. By default, the Router Advertisement MTU option is not suppressed.

Example

- This command suppresses the MTU option on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-vl200)#ipv6 nd ra mtu suppress
switch(config-vl200)#
```

Router Advertisment Flag Configuration

The **ipv6 nd managed-config-flag** command configures the switch to set the ***managed address configuration*** flag in IPv6 router advertisements transmitted from the configuration mode interface. This bit instructs receptive hosts to use stateful address autoconfiguration.

The **ipv6 nd other-config-flag** command configures the switch to set the ***other stateful configuration*** flag in IPv6 router advertisements transmitted from the configuration mode interface. This flag indicates the availability of autoconfiguration information, other than addresses, and that hosts should use stateful autoconfiguration to obtain the information. If the ***managed address configuration*** flag is set, then an attached host can use stateful autoconfiguration to obtain the other (nonaddress) information regardless of the setting of the ***other stateful configuration*** flag.

Examples

- These commands configure the switch to set the managed address configuration flag in IPv6 router advertisements sent from VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd managed-config-flag
switch(config-if-Vl200)#
```

- These commands configure the switch to set the other stateful configuration flag in IPv6 router advertisements sent from VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd other-config-flag
switch(config-if-Vl200)#
```

18.3.2.4 IPv6 Router Preference

The IPv6 Router Preference protocol supports an extension to Router Advertisement messages for communicating default router preferences and more specific routes from routers to hosts. This provides assistance to hosts when selecting a router. RFC 4191 describes the IPv6 Router Preference Protocol.

The **ipv6 nd router-preference** command specifies the value that the switch enters in the default router preference (DRP) field of router advertisements (RAs) that it sends from the configuration mode interface. The default field entry value is ***medium***.

IPv6 hosts maintain a default router list from which it selects a router for traffic to offlink destinations, then caches the router's address in the destination cache. The neighbor discovery protocol (NDP) prefers routers that are ***reachable*** or ***probably reachable*** over routers whose reachability is unknown or suspect. For ***reachable*** or ***probably reachable*** routers, NDP can either select the same router every time or cycle through the router list. DRP values specify a host's preferred router.

Example

- This command configures the switch as a medium preference router on RAs sent from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd router-preference medium
switch(config-if-Vl200)#
```

18.3.3 Viewing IPv6 Network Components

Displaying the FIB and Routing Table

The **show ipv6 route** command displays routing table entries that are in the forwarding information base (FIB), including static routes, routes to directly connected networks, and dynamically learned routes. Multiple equal-cost paths to the same prefix are displayed contiguously as a block, with the destination prefix displayed only on the first line.

Example

- This command displays a route table entry for a specific IPv6 route.

```
switch>show ipv6 route fd7a:3418:52a4:fe18::/64
IPv6 Routing Table - 77 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP,
A - Aggregate

O   fd7a:3418:52a4:fe18::/64 [10/20]
     via f180::21c:73ff:fe00:1319, Vlan3601
     via f180::21c:73ff:fe00:1319, Vlan3602
     via f180::21c:73ff:fe00:1319, Vlan3608
     via f180::21c:73ff:fe0f:6a80, Vlan3610
     via f180::21c:73ff:fe00:1319, Vlan3611

switch>
```

Displaying Host Routes

The **show ipv6 route host** command displays all host routes in the IPv6 host forwarding table. Host routes are those whose destination prefix is the entire address (prefix = /128). Each displayed host route is labeled with its purpose:

— F static routes from the FIB.
— R routes defined because the IP address is an interface address.
— A routes to any neighboring host for which the switch has an ARP entry.

Example

- This command displays all IPv6 host routes in the host forwarding table.

```
switch#show ipv6 route host
R - receive F - FIB, A - attached

 F  ::1 to cpu
 A  fee7:48a2:0c11:1900:400::1 on Vlan102
 R  fee7:48a2:0c11:1900:400::2 to cpu
 F  fee7:48a2:0c11:1a00::b via fe80::21c:73ff:fe0b:a80e on Vlan3902
 R  fee7:48a2:0c11:1a00::17 to cpu
 F  fee7:48a2:0c11:1a00::20 via fe80::21c:73ff:fe0b:33e on Vlan3913
 F  fee7:48a2:0c11:1a00::22 via fe80::21c:73ff:fe01:5fe1 on Vlan3908
                            via fe80::21c:73ff:fe01:5fe1 on Vlan3902

switch#
```

18.4 IPv6 Command Descriptions

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Commands

- ipv6 hardware fib aggregate-address . . . . . Page 740
- ipv6 neighbor . . . . . Page 751
- ipv6 route . . . . . Page 752
- ipv6 unicast-routing . . . . . Page 753

Interface Configuration Commands

- ipv6 address . . . . . Page 738
- ipv6 enable . . . . . Page 739
- ipv6 nd managed-config-flag . . . . . Page 741
- ipv6 nd other-config-flag . . . . . Page 742
- ipv6 nd prefix . . . . . Page 743
- ipv6 nd ra interval . . . . . Page 745
- ipv6 nd ra lifetime . . . . . Page 746
- ipv6 nd ra mtu suppress . . . . . Page 747
- ipv6 nd ra suppress . . . . . Page 748
- ipv6 nd reachable-time . . . . . Page 749
- ipv6 nd router-preference . . . . . Page 750

Privileged EXEC Commands

- clear ipv6 neighbors . . . . . Page 737

EXEC Commands

- show ipv6 hardware fib aggregate-address . . . . . Page 754
- show ipv6 interface . . . . . Page 755
- show ipv6 nd ra internal state . . . . . Page 756
- show ipv6 neighbor . . . . . Page 757
- show ipv6 route . . . . . Page 758
- show ipv6 route host . . . . . Page 759
- show ipv6 route interface . . . . . Page 760
- show ipv6 route tag . . . . . Page 761

clear ipv6 neighbors

The **clear ipv6 neighbors** command removes the specified dynamic IPv6 neighbor discovery cache entries. Commands that do not specify an IPv6 address remove all dynamic entries for the listed interface. Commands that do not specify an interface remove all dynamic entries.

Command Mode

Privileged EXEC

Command Syntax

clear ipv6 neighbors [*PORT*] [*DYNAMIC_IPV6*]

Parameters

- ***PORT*** Interface through which neighbor is accessed. Options include:
  - <no parameter> all dynamic entries are removed.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
- ***DYNAMIC_IPV6*** Address of entry removed by the command. Options include:
  - <no parameter> all dynamic entries for specifed interface are removed.
  - *ipv6_addr* IPv6 address of entry.

Example

- This command removes all dynamic neighbor entries for VLAN interface 200.

```
switch#clear ipv6 neighbors vlan 200
switch#
```

ipv6 address

The **ipv6 address** command enables IPv6 processing on the configuration mode interface, assigns an IPv6 address to the interface, and defines the prefix length for the associated address space. An interface can have multiple IPv6 address assignments.

The **no ipv6 address** and **default ipv6 address** commands remove the IPv6 address assignment from the configuration mode interface by deleting the corresponding **ipv6 address** command from ***running-config***. If the command does not include an address, all address assignments are removed from the interface. IPv6 processing remains enabled on the interface after the removal of all IPv6 addresses only if an **ipv6 enable** command is configured on the interface.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 address ipv6_prefix
no ipv6 address [ipv6_prefix]
default ipv6 address [ipv6_prefix]
```

Parameters

- *ipv6_prefix* address assigned to the interface (CIDR notation).

Guidelines

This command is supported on routable interfaces.

Example

- These commands configure an IPv6 address and prefix length for VLAN 200:

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 address 10:23:31:00:01:32:93/64
switch(config-if-Vl200)#
```

ipv6 enable

The **ipv6 enable** command enables IPv6 processing on the configuration mode interface. Assigning an IPv6 address to an interface also enables IPv6 processing on the interface.

The **no ipv6 enable** and **default ipv6 enable** command remove the corresponding **ipv6 enable** command from ***running-config***. This action disables IPv6 processing on interfaces that are not configured with an IPv6 address.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 enable
no ipv6 enable
default ipv6 enable
```

Example

- This command enables IPv6 processing on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-vl200)#ipv6 enable
switch(config-vl200)#
```

ipv6 hardware fib aggregate-address

The **ipv6 hardware fib aggregate-address** command specifies the routing table repository of specified IPv6 route.

By default, routes that are created statically through the CLI or dynamically through routing protocols are initially stored in software routing tables, then entered in the hardware routing table by the routing agent. This command prevents the entry of the specified route into the hardware routing table. Specified routes that are in the hardware routing table are removed by this command. Specific routes that are encompassed within the specified route prefix are affected by this command.

The **no ipv6 hardware fib aggregate-address** and **default ipv6 hardware fib aggregate-address** commands remove the restriction from the hardware routing table for the specified routes by removing the corresponding **ipv6 hardware fib aggregate-address** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ipv6 hardware fib aggregate-address ipv6_prefix summary-only software-forward
no ipv6 hardware fib aggregate-address ipv6_prefix
default ipv6 hardware fib aggregate-address ipv6_prefix
```

Parameters

- *ipv6_prefix* IPv6 prefix that is restricted from the hardware routing table (CIDR notation).

Example

- These commands configure a hardware routing restriction for an IPv6 prefix, then displays that restriction.

```
switch(config)#ipv6 hardware fib aggregate-address fd77:4890:5313:ffed::/64
summary-only software-forward
switch(config)#show ipv6 hardware fib aggregate-address
Codes: S - Software Forwarded
S  fd77:4890:5313:ffed::/64

switch(config)#
```

ipv6 nd managed-config-flag

The **ipv6 nd managed-config-flag** command configures the switch to set the ***managed address configuration*** flag in IPv6 router advertisements transmitted from the configuration mode interface.

The **no ipv6 nd managed-config-flag** and **default ipv6 nd managed-config-flag** commands restore the default setting where the ***managed address configuration*** flag is not set in IPv6 router advertisements from the interface by removing the corresponding **ipv6 nd managed-config-flag** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd managed-config-flag
no ipv6 nd managed-config-flag
default ipv6 nd managed-config-flag
```

Example

- These commands configure the switch to set the managed address configuration flag in IPv6 router advertisements sent from VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd managed-config-flag
switch(config-if-Vl200)#
```

ipv6 nd other-config-flag

The **ipv6 nd other-config-flag** command configures the switch to set the ***other stateful configuration*** flag in IPv6 router advertisements transmitted from the configuration mode interface.

The ***other stateful configuration*** flag indicates the availability of autoconfiguration information, other than addresses. When the flag is set, attached hosts need to use stateful autoconfiguration to obtain the information. If the ***managed address configuration*** flag is set (**ipv6 nd managed-config-flag**), then an attached host can use stateful autoconfiguration to obtain the other (nonaddress) information regardless of the setting of the "other stateful configuration" flag.

The **no ipv6 nd other-config-flag** and **default ipv6 nd other-config-flag** commands restore the default setting where the ***other stateful configuration*** flag is not set in IPv6 router advertisements from the interface by removing the corresponding **ipv6 nd other-config-flag** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd other-config-flag
no ipv6 nd other-config-flag
default ipv6 nd other-config-flag
```

Example

- These commands configure the switch to set the other stateful configuration flag in IPv6 router advertisements sent from VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd other-config-flag
switch(config-if-Vl200)#
```

ipv6 nd prefix

The **ipv6 nd prefix** command configures neighbor discovery router advertisement prefix inclusion for router advertisements (RAs) sent from the configuration mode interface.

By default, all prefixes configured as IPv6 addresses (**ipv6 address**) are advertised in the interface's RAs. The **ipv6 nd prefix** command with the ***no-advertise*** option prevents advertising of the specified prefix without affecting the advertising of other prefixes specified as IPv6 addresses. When an interface configuration includes at least one **ipv6 nd prefix** command that enables prefix advertising, RAs advertise only prefixes specified through **ipv6 nd prefix** commands.

Commands enabling prefix advertising also specify the advertised valid and preferred lifetime periods. Default periods are 2,592,000 (valid) and 604,800 (preferred) seconds.

The **no ipv6 nd prefix** and **default ipv6 nd prefix** commands remove the corresponding **ipv6 nd prefix** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd prefix ipv6_prefix LIFETIME [FLAGS]
ipv6 nd prefix ipv6_prefix no-advertise
no ipv6 nd prefix ipv6_prefix
default ipv6 nd prefix ipv6_prefix
```

Parameters

- *ipv6_prefix* IPv6 prefix (CIDR notation).
- **no-advertise** Prevents advertising of the specified prefix.
- ***LIFETIME*** Period that the specified IPv6 prefix is advertised (seconds). Options include
  - — *valid preferred* Two values that set the ***valid*** and ***preferred*** lifetime periods.
  - — *valid* One value that sets the ***valid*** lifetime. The ***preferred*** lifetime is set to the default value.
  - — <no parameter> The ***valid*** and ***preferred*** lifetime periods are set to their default values.

  Options for *valid*: <***0*** to ***4294967295***> and **infinite**. Default value is 2592000
  Options for *preferred*: <***0*** to ***4294967295***> and **infinite**. Default value is 604800
  The maximum value (***4294967295***) and **infinite** are equivalent settings.
- ***FLAGS*** ***on-link*** and ***autonomous address-configuration*** flag values in router advertisements.
  - — <no parameter> both flags are set.
  - — **no-autoconfig** ***autonomous address-configuration*** flag is reset.
  - — **no-onlink** ***on-link*** flag is reset.
  - — **no-autoconfig no-onlink** both flags are reset.
  - — **no-onlink no-autoconfig** both flags are reset.

Example

- These commands enable neighbor discovery advertising for IPv6 address 3012:D678::/64, on VLAN interface 200, specifying a valid lifetime of 1,296,000 seconds and the default preferred lifetime.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd prefix 3012:D678::/64 1296000
switch(config-if-Vl200)#
```

ipv6 nd ra interval

The **ipv6 nd ra interval** command configures the interval between IPv6 router advertisement (RA) transmissions from the configuration mode interface.

The **no ipv6 nd ra interval** and **default ipv6 nd ra interval** commands return the IPv6 RA transmission interval for the configuration mode interface to the default value of 200 seconds by removing the corresponding **ipv6 nd ra interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra interval [SCALE] ra_period [minimum_period]
no ipv6 nd ra interval
default ipv6 nd ra interval
```

Parameters

- *SCALE* timescale in which command parameter values are expressed.
  - <no parameter> seconds
  - **msec** milliseconds
- *ra_period* maximum interval between successive IPv6 router advertisement transmissions. The default period is 200 seconds.
  - *<**4 - 1800**>* valid range when **scale** is set to default value (seconds).
  - *<**500 - 1800000**>* valid range when **scale** is set to ***msec***.
- *minimum_period* minimum interval between successive IPv6 router advertisement transmissions. Must be smaller than *ra_period*. By default, a minimum period is not defined.
  - <no parameter> Command does not specify a minimum period.
  - *<**3 - 1799**>* valid range when **scale** is set to default value (seconds).
  - *<**375 - 1799999**>* valid range when **scale** is set to ***msec***.

Example

- These commands configure a router advertisement transmission interval of 60 seconds on VLAN interface 200, then displays the interface status.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra interval 60
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 nd ra interval 60
switch(config-if-Vl200)#
```

ipv6 nd ra lifetime

The **ipv6 nd ra lifetime** command specifies the value that the switch places in the ***router lifetime*** field of IPv6 router advertisements (RAs) sent from the configuration mode interface. The router lifetime specifies the period that the router can be considered as a default router by RA recipients.

Setting the value to 0 indicates that the router should not be considered a default router on this interface. The ***router lifetime*** value can be set to a nonzero value to indicate that it should be considered a default router on this interface. The nonzero value for the ***router lifetime*** value should not be less than the router advertisement interval

The **no ipv6 nd ra lifetime** and **default ipv6 nd ra lifetime** commands return the IPv6 RA lifetime data entry filed for the configuration mode interface to the default value of 1800 seconds by removing the corresponding **ipv6 nd ra lifetime** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra lifetime ra_lifetime
no ipv6 nd ra lifetime
default ipv6 nd ra lifetime
```

Parameters

- *ra_lifetime* router lifetime period (seconds). Default value is 1800. Options include
  - — ***<0>*** Router should not be considered as a default router
  - — ***<1 - 65535>*** Lifetime period advertised in RAs.

Example

- This command configures the switch to enter 2700 in the router lifetime field of RAs tranmitted from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd ra lifetime 2700
switch(config-if-Vl200)#show active
interface Vlan20
   ipv6 nd ra lifetime 2700
switch(config-if-Vl200)#
```

ipv6 nd ra mtu suppress

The **ipv6 nd ra mtu suppress** command suppresses the Router Advertisement MTU option on the configuration mode interface. The MTU option ensures that all nodes on a link use the same MTU value. By default, the Router Advertisement MTU option is not suppressed.

The **no ipv6 nd ra mtu suppress** and **default ipv6 nd ra mtu suppress** commands restores the MTU option setting to enabled by for the configuration mode interface by removing the corresponding **ipv6 nd ra mtu suppress** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra mtu suppress
no ipv6 nd ra mtu suppress
default ipv6 nd ra mtu suppress
```

Example

- This command suppresses the MTU option on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-vl200)#ipv6 nd ra mtu suppress
switch(config-vl200)#
```

ipv6 nd ra suppress

The **ipv6 nd ra suppress** command suppress IPv6 router advertisement transmissions on the configuration mode interface. By default, only unsolicited RAs that are transmitted periodically are suppressed. The **all** option configures the switch to suppress all RAs, including those responding to a router solicitation.

The **no ipv6 nd ra suppress** and **default ipv6 nd ra suppress** commands restore the transmission of RAs on the configuration mode interface by deleting the corresponding **ipv6 nd ra suppress** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd ra suppress [SCOPE]
no ipv6 nd ra suppress
default ipv6 nd ra suppress
```

Parameters

- *SCOPE* specifies the RAs that are suppressed.
  - <no parameter> Periodic unsolicited RAs are suppressed.
  - **all** All RAs are suppressed.

Example

- This command suppresses all RAs on VLAN interface 200.

```
switch(config)#interface vlan 200
switch(config-vl200)#ipv6 nd ra suppress all
switch(config-vl200)#
```

ipv6 nd reachable-time

The **ipv6 nd reachable-time** command specifies the time period that the switch includes in the reachable time field of Router Advertisements (RAs) sent from the configuration mode interface. The reachable time defines the period that a remote IPv6 node is considered reachable after a reachability confirmation event.

RAs that advertise zero seconds indicate that the router does not specify a reachable time. The default advertisement value is 0 seconds. The switch reachability default period is 30 seconds.

The **no ipv6 nd reachable-time** and **default ipv6 nd reachable-time** commands restore the entry of the default value (0) in RAs sent from the configuration mode interface by deleting the corresponding **ipv6 nd reachable-time** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd reachable-time period
no ipv6 nd reachable-time
default ipv6 nd reachable-time
```

Parameters

- *period* Reachable time value (milliseconds). Value ranges from ***0*** to ***4294967295***. Default is **0**.

Example

- These commands configure the entry of 25000 (25 seconds) in the reachable time field of RAs sent from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd reachable-time 25000
interface Vlan200
   ipv6 address fd7a:4321::1/64
   ipv6 nd reachable-time 25000
switch(config-if-Vl200)#
```

ipv6 nd router-preference

The **ipv6 nd router-preference** command specifies the value that the switch enters in the default router preference (DRP) field of router advertisements (RAs) that it sends from the configuration mode interface. The default field entry value is ***medium***.

IPv6 hosts maintain a default router list from which it selects a router for traffic to offlink destinations, then caches the router's address in the destination cache. The neighbor discovery protocol (NDP) prefers routers that are ***reachable*** or ***probably reachable*** over routers whose reachability is unknown or suspect. For ***reachable*** or ***probably reachable*** routers, NDP can either select the same router every time or cycle through the router list. DRP values specify a host's preferred router.

The **no ipv6 nd router-preference** and **default ipv6 nd router-preference** commands restore the switch to enter the default DRP field value of ***medium*** in RAs sent from the configuration mode interface by deleting the corresponding **ipv6 nd router-preference** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Management Configuration
Interface-Port-channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 nd router-preference RANK
no ipv6 nd router-preference
default ipv6 nd router-preference
```

Parameters

- ***RANK*** Router preference value. Options include:
  - **high**
  - **low**
  - **medium**

Example

- This command configures the switch as a medium preference router on RAs sent from VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 nd router-preference medium
switch(config-if-Vl200)#
```

ipv6 neighbor

The **ipv6 neighbor** command creates an IPv6 neighbor discovery cache static entry. The command converts pre-existing dynamic cache entries for the specified address to static entries.

The **no ipv6 neighbor** and **default ipv6 neighbor** commands remove the specified static entry from the IPV6 neighbor discovery cache and delete the corresponding **ipv6 neighbor** command from ***running-config***. These commands do not affect any dynamic entries in the cache.

Command Mode

Global Configuration

Command Syntax

```
ipv6 neighbor ipv6_addr PORT mac_addr
no ipv6 neighbor ipv6_address PORT
default ipv6 neighbor ipv6_addr PORT
```

Parameters

- *ipv6_addr* Neighbor's IPv6 address.
- ***PORT*** Interface through which the neighbor is accessed. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
- *mac_addr* Neighbor's data-link (hardware) address. (48-bit dotted hex notation – H.H.H).

Example

- This command adds a static entry to the neighbor discovery cache for the neighbor located at 3100:4219::3EF2 with hardware address 0100.4EA1.B100 and accessible through VLAN 200.

```
switch(config)#ipv6 3100:4219::3EF2 vlan 200 0100.4EA1.B100
switch(config)#
```

ipv6 route

The **ipv6 route** command creates an IPv6 static route. The destination is a IPv6 prefix; the source is an IPv6 address or a routable interface port. When multiple routes exist to a destination prefix, the route with the lowest administrative distance takes precedence.

Static routes have a default administrative distance of 1. Assigning a higher administrative distance to a static route configures it to be overridden by dynamic routing data. For example, a static route with a distance value of 200 is overridden by OPSF intra-area routes with a default distance of 110.

Multiple routes that are configured to the same destination with the same administrative distance comprise an Equal Cost Multi-Path (ECMP) route. The switch attempts to spread outbound traffic to all ECMP route paths equally. All ECMP paths are assigned the same tag value; commands that change the tag value of any ECMP path change the tag value of all paths in the ECMP.

The **no ipv6 route** and **default ipv6 route** commands delete a static route by removing the corresponding **ipv6 route** command from ***running-config***. If the command includes no parameters, all configured routes to the destination are deleted. If an argument is specified, only routes with matching parameters are deleted. Parameters that are not in the command line are not evaluated.

Command Mode

Global Configuration

Command Syntax

```
ipv6 route dest_prefix SOURCE [distance] [TAG_OPTION]
no ipv6 route dest_prefix [SOURCE] [distance]
default ipv6 route dest_prefix [SOURCE] [distance]
```

Parameters

- *dest_prefix* destination IPv6 prefix (CIDR notation).
- ***SOURCE*** Route source. Options include:
  - *ipv6_addr* IPv6 address.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **null0** Null0 interface.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
- *distance* administrative distance assigned to route. Options include:
  - <no parameter> route assigned default administrative distance of one.
  - <1-255> The administrative distance assigned to route.
- ***TAG_OPTION*** static route tag. Options include:
  - <no parameter> assigns default static route tag of 0.
  - **tag** *t_value* Static route tag value. *t_value* ranges from ***0*** to ***4294967295***.

Example

- This command creates an IPv6 static route.

```
switch(config)#ipv6 route 10:23:31:00:01:32:93/24 vlan 300
switch(config)#
```

ipv6 unicast-routing

The **ipv6 unicast-routing** command enables the forwarding of IPv6 unicast packets. When routing is enabled, the switch attempts to deliver inbound packets to destination addresses by forwarding them to interfaces or next hop addresses specified by the IPv6 routing table.

The **no ipv6 unicast-routing** and default **ip ipv6 unicast-routing** commands disable IPv6 unicast routing by removing the **ipv6 unicast-routing** command from *running-config*. Dynamic routes added by routing protocols are removed from the routing table. Static routes are preserved by default; the **delete-static-routes** option removes static entries from the routing table.

IPv6 unicast routing is disabled by default.

Command Mode

Global Configuration

Command Syntax

```
ipv6 unicast-routing
no ipv6 unicast-routing [DELETE_ROUTES]
default ipv6 unicast-routing [DELETE_ROUTES]
```

Parameters

- *DELETE_ROUTES* Resolves routing table static entries when routing is disabled.
  - <no parameter> Routing table retains static entries.
  - **delete-static-routes** Static entries are removed from the routing table.

Example

- This command enables IPv6 unicast-routing.

```
switch(config)#ipv6 unicast-routing
switch(config)#
```

show ipv6 hardware fib aggregate-address

The **show ipv6 hardware fib aggregate-address** command displays the IPv6 prefixes that are restricted from entry into the hardware routing table. The **ipv6 hardware fib aggregate-address** command configures IPv6 prefix restictions.

Command Mode

EXEC

Command Syntax

show ipv6 address fib aggregate-address [***ADDRESS***] [***RESTRICTION***]

Parameters

- ***ROUTE_FILTER*** filters by IPv6 address. Options include:
  - <no parameter> Displays all routes.
  - *ipv6_addr* Command displays only specified address.
  - *ipv6_prefix* Command displays addresses filtered by specifed prefix (CIDR notation).
- ***RESTRICTION*** filters by route restriction.
  - <no parameter> displays routes restricted from the hardware routing table.
  - **software-forward** displays routes restricted from the hardware routing table.

Example

- This command displays the routes that are restricted from the hardware routing table.

```
switch>show ipv6 hardware fib aggregate-address
Codes: S - Software Forwarded
S  fd77:4890:5313:aaed::/64
S  fd77:4890:5313:ffed::/64

switch>
```

show ipv6 interface

The **ipv6 interface** command displays the status of specified routed interfaces that are configured for IPv6.

Command Mode

EXEC

Command Syntax

show ipv6 interface [***INTERFACE_NAME***]

Parameters

- *INTERFACE_NAME* interfaces for which command displays status.
  - <no parameter> all routed interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command displays the status of VLAN 903.

```
switch>show ipv6 interface vlan 903
Vlan903 is up, line protocol is up (connected)
  IPv6 is enabled, link-local is fe80::21c:73ff:fe01:21e/64
  Global unicast address(es):
    fd7a:629f:52a4:fe10::3, subnet is fd7a:629f:52a4:fe10::/64
  Joined group address(es):
    ff02::1
    ff02::1:ff01:21e
    ff02::1:ff00:3
    ff01::2
switch>
```

show ipv6 nd ra internal state

The **ipv6 nd ra internal state** command displays the state of the IPv6 Router Advertisement daemon for the specified routable interface.

Command Mode
EXEC

Command Syntax
show ipv6 nd ra internal state [***INTERFACE_NAME***]

Parameters

- *INTERFACE_NAME* interfaces for which command displays status.
  - <no parameter> all routed interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command displays the IPv6 RA daemon for VLAN interface 1243.

```
switch>show ipv6 nd ra internal state vlan 1243
INTERFACE: Vlan3908
                                  ifindex : 0x00000021
                                      mtu : 9212
                                numIpv6Addr : 2
                       numPrefixToAdvertise : 0
                        numPrefixToSuppress : 0
                                 RaSuppress : 0
                              RsRspSuppress : 0
                           raIntervalMaxMsec : 200000
                           raIntervalMinMsec : 0
                           managedConfigFlag : 0
                             otherConfigFlag : 0
                               raMtuSuppress : 0
                                  raLifetime : 1800
                              reacheableTime : 0
                           routerPreference : 0
                                 lastRaTime : 2012-05-01 09:22:57.020634
                          lastRsRspSentTime :
                                nextTimeout : 171.474535 (sec)
                    raNotSentIntfNotReady : 0
                                numRaSent : 219
                                numRsRcvd : 0
                             numRsSuppressed : 0
                               numRsRspSent : 0
               numRsDroppedInvalidHopLimit : 0
               numPktDroppedUnexpectedType : 0
                               initialized : 1
switch>
```

show ipv6 neighbor

The **show ipv6 neighbor** command displays the IPv6 neighbor discovery cache. The command provides filters to restrict the list to a specified IPv6 address or routable interface.

Command Mode

EXEC

Command Syntax

show ipv6 neighbor [*PORT*] [*SOURCE*]

Parameters

- *PORT* Filters by interface through which neighbor is accessed. Options include:
  - <no parameter> all routed interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
- *PORT* Filters by neighbor IPv6 address. Options include:
  - <no parameter> all IPv6 neighbors.
  - *ipv6_addr* IPv6 address of individual neighbor.

Example

- This command displays the IPv6 neighbor discovery cache for IPv6 address ***fe80::21c:73ff:fe01:5fe1***.

```
switch>show ipv6 neighbor fe80::21c:73ff:fe01:5fe1
IPv6 Address                              Age Hardware Addr    State Interface
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Et12
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Po999
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl102
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl103
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl205
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl207
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl3901
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl3902
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl3903
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl3904
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl3905
fe80::21c:73ff:fe01:5fe1                    0 001c.d147.8214   REACH Vl3996
```

show ipv6 route

The **show ipv6 route** command displays IPv6 routing table entries that are in the forwarding information base (FIB), including static routes, routes to directly connected networks, and dynamically learned routes. Multiple equal-cost paths to the same prefix are displayed contiguously as a block, with the destination prefix displayed only on the first line.

The **show running-config** command displays all configured routes.

Command Mode
EXEC

Command Syntax
show ipv6 route [***ADDRESS***] [***ROUTE_TYPE***] [***INFO_LEVEL***]

Parameters

Address, when present, is always listed first. All other parameters can be placed in any order.

- ***ADDRESS*** filters routes by IPv6 address or prefix.
  - <no parameter> all routing table entries.
  - *ipv6_address* routing table entries matching specified IPv6 address.
  - *ipv6_prefix* routing table entries matching specified IPv6 prefix (CIDR notation).
- ***ROUTE_TYPE*** filters routes by specified protocol or origin.
  - <no parameter> all routing table entries.
  - **aggregate** entries for BGP aggregate routes.
  - **bgp** entries added through BGP protocol.
  - **connected** entries for routes to networks directly connected to the switch.
  - **kernel** entries appearing in Linux kernel but not added by EOS software.
  - **ospf** entries added through OSPF protocol.
  - **static** entries added through CLI commands.
- ***INFO_LEVEL*** Filters entries by next hop connection.
  - <no parameter> filters routes whose next hops are directly connected.
  - **detail** displays all routes.

Example

- This command displays a route table entry for a specific IPv6 route.

```
switch>show ipv6 route fd7a:3418:52a4:fe18::/64
IPv6 Routing Table - 77 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

O   fd7a:3418:52a4:fe18::/64 [10/20]
     via fe80::21c:73ff:fe00:1319, Vlan3601
     via fe80::21c:73ff:fe00:1319, Vlan3602
     via fe80::21c:73ff:fe00:1319, Vlan3608
     via fe80::21c:73ff:fe0f:6a80, Vlan3610
     via fe80::21c:73ff:fe00:1319, Vlan3611

switch>
```

show ipv6 route host

The **show ipv6 route host** command displays all host routes in the IPv6 host forwarding table. Host routes are those whose destination prefix is the entire address (prefix = /128). Each displayed host route is labeled with its purpose:

- F static routes from the FIB.
- R routes defined because the IP address is an interface address.
- A routes to any neighboring host for which the switch has an ARP entry.

Command Mode

EXEC

Command Syntax

```
show ipv6 route host
```

Example

- This command displays all IPv6 host routes in the host forwarding table.

```
switch>show ipv6 route host
R - receive F - FIB, A - attached

 F  ::1 to cpu
 A  fee7:48a2:0c11:1900:400::1 on Vlan102
 R  fee7:48a2:0c11:1900:400::2 to cpu
 F  fee7:48a2:0c11:1a00::b via fe80::21c:73ff:fe0b:a80e on Vlan3902
 R  fee7:48a2:0c11:1a00::17 to cpu
 F  fee7:48a2:0c11:1a00::20 via fe80::21c:73ff:fe0b:33e on Vlan3913
 F  fee7:48a2:0c11:1a00::22 via fe80::21c:73ff:fe01:5fe1 on Vlan3908
                            via fe80::21c:73ff:fe01:5fe1 on Vlan3902

switch>
```

show ipv6 route interface

The **show ipv6 route interface** command displays routing table entries on a specified routed port.

Command Mode
EXEC

Command Syntax
show ipv6 route [***ADDRESS***] **interface** ***PORT_NAME*** [***INFO_LEVEL***]

Parameters

ADDRESS, when present, is always listed first. All other parameters can be placed in any order.

- ***ADDRESS*** filters routes by IPv6 address or prefix.
  - <no parameter> all routing table entries.
  - *ipv6_address* routing table entries matching specified IPv6 address.
  - *ipv6_prefix* routing table entries matching specified IPv6 prefix (CIDR notation).
- ***PORT_NAME*** interfaces for which command displays status.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
- ***INFO_LEVEL*** Filters entries by next hop connection.
  - <no parameter> filters routes whose next hops are directly connected.
  - **detail** displays all routes.

Example

- This command displays the IPv6 routes in VLAN interface 661.

```
switch>show ipv6 route interface ethernet 8
IPv6 Routing Table - 77 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

O   fd7a:629f:63af:1232::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
O   fd7a:629f:63af:4118::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
O   fd7a:629f:63af:4119::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
O   fd7a:629f:63af:411a::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
O   fd7a:629f:63af:fe78::/64 [150/11]
     via fe80::823c:73ff:fe00:3640, Ethernet8
C   fd7a:629f:63af:fe88::/64 [0/1]
     via ::, Ethernet12
O   fd7a:629f:63af:fe8c::/64 [10/20]
     via fe80::21c:73ff:fe00:3640, Ethernet8
C   fe80:0:40::/64 [0/1]
     via ::, Ethernet8
```

show ipv6 route tag

The **show ipv6 route tag** command displays the route tag assigned to the specified IPv6 address or prefix. Route tags are added to static routes for use by route maps.

Command Mode

EXEC

Command Syntax

show ipv6 route *ADDRESS* tag

Parameters

- *ADDRESS* filters routes by IPv6 address or prefix.
  - *ipv6_address* routing table entries matching specified address (A:B:C:D:E:F:G:H)
  - *ipv6_prefix* routing table entries matching specified IPv6 prefix (A:B:C:D:E:F:G:H/PL).

Example

- This command displays the route tag for the specified prefix.

```
switch>show ipv6 route fd7b:789f:5314:fe08::/64 tag
IPv6 Routing Table - 74 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP, A -
Aggregate

C   fd7b:789f:5314:fe08::/64 tag 0

switch>
```

Chapter 19

Open Shortest Path First – Version 2

Open Shortest Path First (OSPF) is a link-state routing protocol that operates within a single autonomous system. OSPF version 2 is defined by RFC 2328.

This chapter contains the following sections.

- Section 19.1: OSPFv2 Introduction
- Section 19.2: OSPFv2 Conceptual Overview
- Section 19.3: Configuring OSPFv2
- Section 19.4: OSPFv2 Examples
- Section 19.5: OSPFv2 Commands

19.1 OSPFv2 Introduction

19.1.1 Supported Features

Arista switches support these OSPFv2 functions:

- A single OSPFv2 instance
- Intra- and inter-area routing
- Type 1 and 2 external routing
- Broadcast and P2P interfaces
- Stub areas
- Not so stubby areas (NSSA) (RFC 3101)
- MD5 Authentication
- Redistribution of static, IP, and BGP routes into OSPFv2 with route map filtering
- Opaque LSAs (RFC 2370)
- Largely industry standard compatible CLI

19.1.2 Features Not Supported

These OSPFv2 functions are not supported in the current version:

- NBMA, demand circuit, and P2MP interfaces
- Graceful restart (RFC 3623)
- OSPFv2 MIB support

19.2 OSPFv2 Conceptual Overview

19.2.1 Storing Link States

OSPFv2 is a dynamic, link-state routing protocol, where links represent interfaces or routable paths. Dynamic routing protocols calculate the most efficient path between locations based on bandwidth and device status.

A link state advertisement (LSA) is an OSPFv2 packet that communicates a router's topology to other routers. The link state database (LSDB) stores an area's topology database and is composed of LSAs received from other routers. Routers update the LSDB by storing LSAs from other routers.

19.2.2 Topology

An autonomous system (AS) is the IP domain where a dynamic protocol routes traffic. In OSPFv2, an AS is composed of areas, which define the LSDB computation boundaries. All routers in an area store identical LSDBs. Routers in different areas exchange updates without storing the entire database, reducing information maintenance on large, dynamic networks.

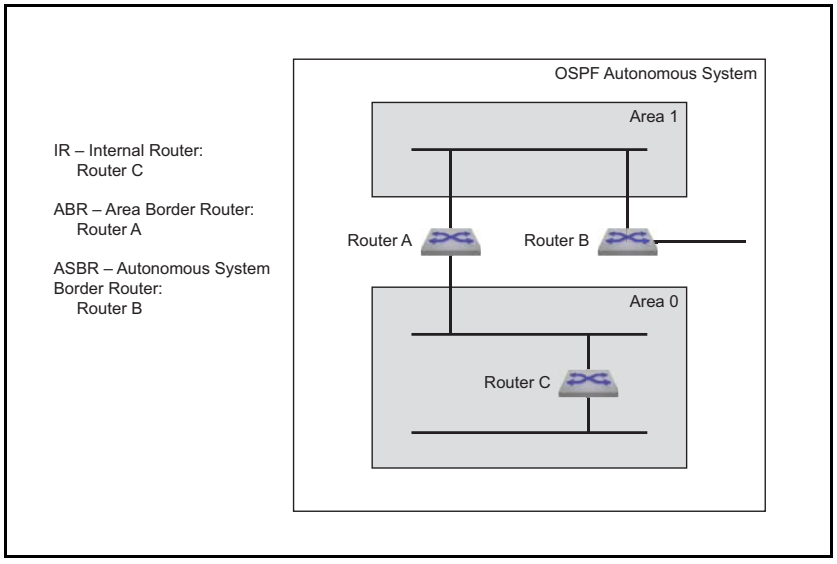
An AS shares internal routing information from its areas and external routing information from other processes to inform routers outside the AS about routes the network can access. Routers that advertise routes on other ASs commit to carry data to the IP space on the route.

OSPFv2 defines these routers:

- Internal router (IR) – a router whose interfaces are contained in a single area. All IRs in an area maintain identical LSDBs.
- Area border router (ABR) – a router that has interfaces in multiple areas. ABRs maintain one LSDB for each connected area.
- Autonomous system boundary router (ASBR) – a gateway router connecting the OSPFv2 domain to external routes, including static routes and routes from other autonomous systems.

Figure 19-1 displays the OSPFv2 router types.

Figure 19-1 OSPFv2 Router Types



OSPFv2 areas are assigned a number between 0 and 4,294,967,295 ($2^{32} – 1$). Area numbers are often expressed in dotted decimal notation, similar to IP addresses.

Each AS has a backbone area, designated as area 0, that connects to all other areas. The backbone receives routing information from all areas, then distributes it to the other areas as required.

OSPFv2 area types include:

- Normal area – accepts intra-area, inter-area, and external routes. The backbone is a normal area.
- Stub area – does not receive router advertisements external to the AS. Stub area routing is based on a default route.
- Not-so-stubby-area (NSSA) – may import external routes from an ASBR, does not receive external routes from the backbone, and does not propagate external routes to other areas.

19.2.3 Link Updates

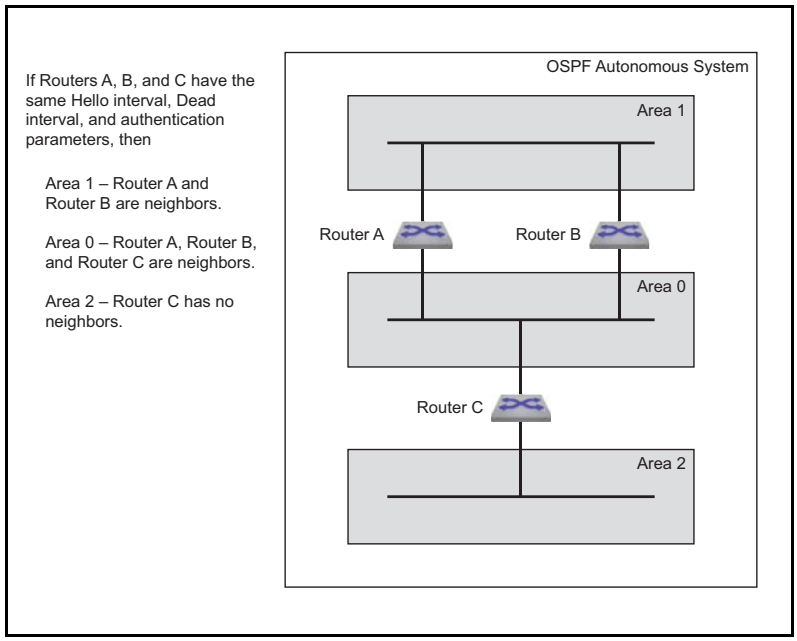
Routers periodically send hello packets to advertise status and establish neighbors. A router's hello packet includes IP addresses of other routers from which it received a hello packet within the time specified by the router dead interval. Routers become neighbors when they detect each other in their hello packets if they:

- share a common network segment.
- are in the same area.
- have the same hello interval, dead interval, and authentication parameters.

Neighbors form adjacencies to exchange LSDB information. A neighbor group uses hello packets to elect a Designated Router (DR) and Backup Designated Router (BDR). The DR and BDR become adjacent to all other neighbors, including each other. Only adjacent neighbors share database information.

Figure 19-2 illustrates OSPFv2 neighbors.

Figure 19-2 OSPFv2 Neighbors



The DR is the central contact for database exchanges. Switches send database information to their DR, which relays the information to the other neighbors. All routers in an area maintain identical LSDBs. Switches also send database information to their BDR, which stores this data without distributing it. If the DR fails, the BDR distributes LSDB information to its neighbors.

OSPFv2 routers distribute LSAs by sending them on all of their active interfaces. Passive interfaces send LSAs to active interfaces but do not receive LSAs, thus alerting OSPFv2 routers of devices that do not otherwise participate in OSPFv2. The router does not send or process OSPFv2 packets received on passive interfaces. including hello packets, which causes the interface to drop its adjacencies.

When a router's LSDB is changed by an LSA, it sends the changes to the DR and BDR for distribution to the other neighbors. Routing information is updated only when the topology changes.

Routing devices use Dijkstra's algorithm to calculate the shortest path to all known destinations, based on cumulative route cost. The cost of an interface indicates the transmission overhead and is usually inversely proportional to its bandwidth.

19.3 Configuring OSPFv2

These sections describe basic OSPFv2 configuration steps:

- Section 19.3.1: Configuring the OSPFv2 Instance
- Section 19.3.2: Configuring OSPFv2 Areas
- Section 19.3.3: Configuring Interfaces for OSPFv2
- Section 19.3.4: OSPFv2 Enabling
- Section 19.3.5: Displaying OSPFv2 Status

19.3.1 Configuring the OSPFv2 Instance

19.3.1.1 Entering OSPFv2 Configuration Mode

OSPFv2 configuration commands apply to the OSPFv2 instance. To perform OSPFv2 configuration commands, the switch must be in router-ospf configuration mode. The **router ospf** command places the switch in router-ospf configuration mode and creates an OSPFv2 instance if one was not previously created.

The switch supports one OSPFv2 instance. When an OSPFv2 instance exists, the **router ospf** command must specify its process ID. Attempts to define additional instances will generate errors.

The process ID identifies the OSPFv2 process of the instance. The process ID is local to the router. Neighbor OSPFv2 routers can have different process IDs.

Example

- This command places the switch in router-ospf configuration mode and, if not previously created, creates an OSPFv2 instance with a process ID of 100.

```
Switch(config)#router ospf 100
Switch(config-router-ospf)#
```

19.3.1.2 Defining the Router ID

The router ID is a 32-bit number assigned to a router running OSPFv2. This number uniquely labels the router within an Autonomous System. Status commands identify the switch through the router ID.

The switch sets the router ID to the first available alternative in the following list:

1. The **router-id** command.
2. The loopback IP address, if a loopback interface is active on the switch.
3. The highest IP address on the router.

The **router-id (OSPFv2)** command configures the router ID for an OSPFv2 instance.

Example

- This command assigns 15.1.1.1 as the OSPFv2 router ID.

```
Switch(config-router-ospf)#router-id 15.1.1.1
Switch(config-router-ospf)#
```

19.3.1.3 Global OSPFv2 Parameters

These router-ospf configuration mode commands define OSPFv2 behavior.

LSA Overload

The **max-lsa (OSPFv2)** command specifies the maximum number of LSAs allowed in an LSDB database and configures the switch behavior when the limit is approached or exceeded. An LSA overload condition triggers these actions:

- Warning: The switch logs OSPF MAXLSAWARNING if the LSDB contains a specified percentage of the LSA maximum.
- Temporary shutdown: When the LSDB exceeds the LSA maximum, OSPFv2 is disabled and does not accept or acknowledge new LSAs. The switch re-starts OSPFv2 after a specified period.
- Permanent shutdown: The switch permanently disables OSPFv2 after performing a specified number of temporary shutdowns. This state usually indicates the need to resolve a network condition that consistently generates excessive LSA packets.

  OSPFv2 is re-enabled with a **router OSPF** command.

The LSDB size restriction is removed by setting the LSA limit to zero.

Example

This command places the OSPFv2 maximum LSA count at 20,000 and configures these actions:

— The switch logs an OSPF MAXLSAWARNING if the LSDB has 8,000 LSAs (40% of 20,000).
— The switch temporarily disables OSPFv2 for 10 minutes if the LSDB contains 20,000 LSAs.
— The switch permanently disables OSPFv2 after four temporary OSPFv2 shutdowns.
— The shutdown counter resets if the LSDB contains less than 20,000 LSAs for 20 minutes.

```
Switch(config-router-ospf)#max-lsa 20000 40 ignore-time 10 ignore-count 4
reset-time 20
Switch(config-router-ospf)#
```

Logging Adjacency Changes

The **log-adjacency-changes (OSPFv2)** command configures the switch to send a syslog message when it detects a link state change or when a neighbor goes up or down.

Examples

- This command configures the switch to send a syslog message when an OSPFv2 neighbor goes up or down.

```
Switch(config-router-ospf)#log-adjacency-changes
Switch(config-router-ospf)#
```

- This command configures the switch to send a syslog message when it detects any link state change.

```
Switch(config-router-ospf)#log-adjacency-changes detail
Switch(config-router-ospf)#
```

Intra-Area Distance

The **distance ospf intra-area (OSPFv2)** command configures the administrative distance for routes contained in a single OSPFv2 area. Administrative distances compare dynamic routes configured by different protocols. The default administrative distance for intra-area routes is 110.

Example

- This command configures an administrative distance of 95 for OSPFv2 intra-area routes.

```
Switch(config-router-ospf)#distance ospf intra-area 95
Switch(config-router-ospf)#
```

Passive Interfaces

The **passive-interface <interface> (OSPFv2)** command prevents the transmission of hello packets on the specified interface. Passive interfaces drop all adjacencies and do not form new adjacencies. Passive interfaces send LSAs but do not receive them. The router does not send or process OSPFv2 packets received on passive interfaces. The router advertises the passive interface in the router LSA.

The **no passive-interface** command re-enables OSPFv2 processing on the specified interface.

Examples

- This command configures VLAN 2 as a passive interface.

```
Switch(config-router-ospf)#passive-interface vlan 2
Switch(config-router-ospf)#
```

- This command configures VLAN 2 as an active interface.

```
Switch(config-router-ospf)#no passive-interface vlan 2
Switch(config-router-ospf)#
```

Redistributing Static Routes

Redistributing static routes causes the OSPFv2 instance to advertise all static routes on the switch as external OSPFv2 routes. The switch does not support redistributing individual static routes.

Example

- The **redistribute (OSPFv2)** command converts the static routes to OSPFv2 external routes.

```
Switch(config-router-ospf)#redistribute static
Switch(config-router-ospf)#
```

- The **no redistribute (OSPFv2)** command stops the advertising of the static routes as OSPFv2 external routes.

```
Switch(config-router-ospf)#no redistribute static
Switch(config-router-ospf)#
```

19.3.2 Configuring OSPFv2 Areas

OSPFv2 areas are configured through area commands. The switch must be in router-ospf configuration mode, as described in Section 19.3.1.1: Entering OSPFv2 Configuration Mode, to run area commands.

Areas are assigned a 32-bit number that is expressed in decimal or dotted-decimal notation. When an OSPFv2 instance spans multiple routers, the switch only configures areas that connect to its interfaces.

19.3.2.1 Configuring the Area Type

The **area (OSPFv2)** command specifies the area type. The switch supports three area types:

- Normal area: Area that accepts intra-area, inter-area, and external routes. The backbone area (area 0) is a normal area.
- Stub area: Area where external routes are not advertised. External routes are reached through a default summary route (0.0.0.0) inserted into stub areas. Networks with no external routes do not require stub areas.
- NSSA (Not So Stubby Area): ASBRs advertise external LSAs directly connected to the area. External routes from other areas are not advertised and are reached through a default summary route.

The default area type is normal.

Examples

- This command configures area 45 as a stub area.

```
Switch(config-router-ospf)#area 45 stub
Switch(config-router-ospf)#
```

- This command configures area 116.92.148.17 as an NSSA.

```
Switch(config-router-ospf)#area 116.92.148.17 NSSA
Switch(config-router-ospf)#
```

19.3.2.2 Assigning Network Segments to the Area

Assigning Routes to an Area

The **network area (OSPFv2)** command assigns the specified network segment to an OSPFv2 area. The network can be entered in CIDR notation or by an address and wildcard mask.

The switch zeroes the host portion of the specified network address; for example. 1.2.3.4/24 converts to 1.2.3.0/24 and 1.2.3.4/16 converts to 1.2.0.0/16

Example

- Each of these equivalent commands assign the network segment 10.1.10.0/24 to area 0.

```
Switch(config-router-ospf)#network 10.1.10.0 0.0.0.255 area 0
Switch(config-router-ospf)#

Switch(config-router-ospf)#network 10.1.10.0/24 area 0
Switch(config-router-ospf)#
```

  In each case, ***running-config*** stores the command in CIDR (prefix) notation.

Summarizing Routes

By default, ABRs create a summary LSA for each route in an area and advertise them to adjacent routers. The **area range (OSPFv2)** command aggregates routing information, allowing the ABR to advertise multiple routes with one LSA. The **area range** command can also suppress route advertisements.

Examples

- Two **network area** commands assign subnets to an area. The **area range** command summarizes the addresses, which the ABR advertises in a single LSA.

```
Switch(config-router-ospf)#network 10.1.25.80 0.0.0.240 area 5
Switch(config-router-ospf)#network 10.1.25.112 0.0.0.240 area 5
Switch(config-router-ospf)#area 5 range 10.1.25.64 0.0.0.192
Switch(config-router-ospf)#
```

- The **network area** command assigns a subnet to an area, followed by an **area range** command that suppresses the advertisement of that subnet.

```
Switch(config-router-ospf)#network 10.12.31.0 0.0.0.255 area 5
Switch(config-router-ospf)#area 5 range 10.12.31.0 0.0.0.255 not-advertise
Switch(config-router-ospf)#
```

19.3.2.3 Configuring Area Parameters

These router-ospf configuration mode commands define OSPFv2 behavior in a specified area.

Default Summary Route Cost

The **area default-cost (OSPFv2)** command specifies the cost of the default summary route that ABRs send into a stub area or NSSA. Summary routes, also called inter-area routes, originate in areas different than their destination.

Example

- This command configures a cost of 15 for the default summary route in area 23.

```
Switch(config-router-ospf)#area 23 default-cost 15
Switch(config-router-ospf)#
```

Filtering Type 3 LSAs

The **area filter (OSPFv2)** command prevents an area from receiving Type 3 (Summary) LSAs from a specified subnet. Type 3 LSAs are sent by ABRs and contain information about one of its connected areas.

Example

- This command prevents the switch from entering Type 3 LSAs originating from the 10.1.1.2/24 subnet into its area 2 LSDB.

```
Switch(config-router-ospf)#area 2 filter 10.1.1.2/24
Switch(config-router-ospf)#
```

19.3.3 Configuring Interfaces for OSPFv2

OSPFv2 interface configuration commands specify transmission parameters for routed ports and SVIs that handle OSPFv2 packets.

19.3.3.1 Configuring Authentication

OSPFv2 authenticates packets through passwords configured on VLAN interfaces. Interfaces connecting to the same area can authenticate packets if they have the same key. By default, OSPFv2 does not authenticate packets.

OSPFv2 supports simple password and message digest authentication:

- Simple password authentication: A password is assigned to an area. Interfaces connected to the area can authenticate packets by enabling authentication and specifying the area password.
- Message digest authentication: Each interface is configured with a key (password) and key-id pair. When transmitting a packet, the interface generates a string, using the MD5 algorithm, based on the OSPFv2 packet, key, and key ID, then appends that string to the packet.

  Message digest authentication supports uninterrupted transmissions during key changes by allowing each interface to have two keys with different key IDs. When a new key is configured on an interface, the router transmits OSPFv2 packets for both keys. The router stops sending duplicate packets when it detects that all of its neighbors are using the new key.

Implementing authentication on an interface is a two step process:

1. Enabling authentication.
2. Configuring a key (password).

To configure simple authentication on a VLAN interface:

Step 1 Enable simple authentication with the **ip ospf authentication** command.

```
switch(config-if-vl12)#ip ospf authentication
```

Step 2 Configure the password with the **ip ospf authentication-key** command.

```
switch(config-if-vl12)#ip ospf authentication-key 0 code123
```

Running-config stores the password as an encrypted string, using a proprietary algorithm.

To configure Message-Digest authentication on a VLAN interface:

Step 1 Enable Message-Digest authentication with the **ip ospf authentication** command.

```
switch(config-if-vl12)#ip ospf authentication message-digest
```

Step 2 Configure the key ID and password with the **ip ospf message-digest-key** command.

```
switch(config-if-vl12)#ip ospf message-digest-key 23 md5 0 code123
```

Running-config stores the password as an encrypted string, using a proprietary algorithm. The key ID (23) is between keywords **message-digest-key** and **md5**.

19.3.3.2 Configuring Intervals

Interval configuration commands determine OSPFv2 packet transmission characteristics for the specified VLAN interface. Interval configuration commands are entered in vlan-interface configuration mode.

Hello Interval

The hello interval specifies the period between consecutive hello packet transmissions from an interface. Each OSPFv2 neighbor should specify the same hello interval, which should not be longer than any neighbor's dead interval.

The **ip ospf hello-interval** command configures the hello interval for the configuration mode interface. The default is 10 seconds.

Example

- This command configures a hello interval of 30 seconds for VLAN 2.

```
Switch(config-if-Vl2)#ip ospf hello-interval 30
Switch(config-if-Vl2)#
```

Dead Interval

The dead interval specifies the period that an interface waits for an OSPFv2 packet from a neighbor before it disables the adjacency under the assumption that the neighbor is down. The dead interval should be configured identically on all OSPFv2 neighbors and be longer than the hello interval of any neighbor.

The **ip ospf dead-interval** command configures the dead interval for the configuration mode interface. The default is 40 seconds.

Example

- This command configures a dead interval of 120 seconds for VLAN 4.

```
Switch(config-if-Vl4)#ip ospf dead-interval 120
Switch(config-if-Vl4)#
```

Retransmit Interval

Routers that send OSPFv2 advertisements to an adjacent router expect to receive an acknowledgment from that neighbor. Routers that do not receive an acknowledgment will retransmit the advertisement. The retransmit interval specifies the period between retransmissions.

The **ip ospf retransmit-interval** command configures the LSA retransmission interval for the configuration mode interface. The default retransmit interval is 5 seconds.

Example

- This command configures a retransmit interval of 15 seconds for VLAN 3.

```
Switch(config-if-Vl3)#ip ospf retransmit-interval 15
Switch(config-if-Vl3)#
```

Transmission Delay

The transmission delay is an estimate of the time that an interface requires to transmit a link-state update packet. OSPFv2 adds this delay to the age of outbound packets to more accurately reflect the age of the LSA when received by a neighbor.

The **ip ospf transmit-delay** command configures the transmission delay for the configuration mode interface. The default transmission delay is one second.

Example

- This command configures a transmission delay of 5 seconds for VLAN 6.

```
Switch(config-if-Vl6)#ip ospf transmit-delay 5
Switch(config-if-Vl6)#
```

19.3.3.3 Configuring Interface Parameters

Interface Cost

The OSPFv2 interface cost (also called metric) reflects the overhead of sending packets across the interface. The cost is typically assigned to be inversely proportional to the bandwidth of the interface.

The **ip ospf cost** command configures the OSPFv2 cost for the configuration mode interface. The default cost is 10.

Example

- This command configures a cost of 15 for VLAN 2.

```
Switch(config-if-Vl2)#ip ospf cost 15
Switch(config-if-Vl2)#
```

Router Priority

Router priority determines preference during designated router (DR) and backup designated router (BDR) elections. Routers with higher priority numbers have preference over other routers. Routers with a priority of zero cannot be elected as a DR or BDR.

The **ip ospf priority** command configures router priority for the configuration mode interface. The default priority is 1.

Examples

- This command configures a router priority of 15 for VLAN 8.

```
Switch(config-if-Vl8)#ip ospf priority 15
Switch(config-if-Vl8)#
```

- This command restores the router priority of 1 for VLAN 7.

```
Switch(config-if-Vl7)#no ip ospf priority
Switch(config-if-Vl7)#
```

19.3.4 OSPFv2 Enabling

19.3.4.1 IPv4 Routing

OSPFv2 requires that IPv4 routing is enabled on the switch. When IP routing is not enabled, entering OSPFv2 configuration mode generates a message.

Example

- This message is displayed if, when entering router-ospf configuration mode, IP routing is not enabled.

```
Switch(config)#router ospf 100
! IP routing not enabled
Switch(config-router-ospf)#
```

- This command enables IP routing on the switch.

```
Switch(config)#ip routing
Switch(config)#
```

19.3.4.2 Disabling OSPFv2

The switch can disable OSPFv2 operations without disrupting the OSPFv2 configuration.

- **shutdown (OSPFv2)** disables all OSPFv2 activity.
- **ip ospf shutdown** disables OSPFv2 activity on a VLAN interface.

The **no shutdown** and **no ip ospf shutdown** commands resume OSPFv2 activity.

Examples

- This command disables OSPFv2 activity on the switch.

```
Switch(config-router-ospf)#shutdown
Switch(config-router-ospf)#
```

- This command resumes OSPFv2 activity on the switch.

```
Switch(config-router-ospf)#no shutdown
Switch(config-router-ospf)#
```

- This command disables OSPFv2 activity on VLAN 5.

```
Switch(config-if-Vl5)#ip ospf shutdown
Switch(config-if-Vl5)#
```

19.3.5 Displaying OSPFv2 Status

This section describes OSPFv2 **show** commands that display OSPFv2 status. General switch methods that provide OSPFv2 information include pinging routes, viewing route status (**show ip route** command), and viewing the configuration (**show running-config** command).

19.3.5.1 OSPFv2 Summary

The **show ip ospf** command displays general OSPFv2 configuration information and operational statistics.

Example

- This command displays general OSPFv2 information.

```
Switch#show ip ospf
Routing Process "ospf 1" with ID 192.168.103.1
 Supports opaque LSA
 Maximum number of LSA allowed 12000
  Threshold for warning message 75%
  Ignore-time 5 minutes, reset-time 5 minutes
  Ignore-count allowed 5, current 0
 It is an area border router
 Hold time between two consecutive SPFs 5000 msecs
 SPF algorithm last executed 00:00:09 ago
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of LSA 27.
 Number of areas in this router is 3. 3 normal 0 stub 0 nssa
    Area BACKBONE(0.0.0.0)
        Number of interfaces in this area is 2
        It is a normal area
        Area has no authentication
        SPF algorithm executed 153 times
        Number of LSA 8. Checksum Sum 0x03e13a
        Number of opaque link LSA 0. Checksum Sum 0x000000
    Area 0.0.0.2
        Number of interfaces in this area is 1
        It is a normal area
        Area has no authentication
        SPF algorithm executed 153 times
        Number of LSA 11. Checksum Sum 0x054e57
        Number of opaque link LSA 0. Checksum Sum 0x000000
    Area 0.0.0.3
        Number of interfaces in this area is 1
        It is a normal area
        Area has no authentication
        SPF algorithm executed 5 times
        Number of LSA 6. Checksum Sum 0x02a401
        Number of opaque link LSA 0. Checksum Sum 0x000000
```

The output lists configuration parameters and operational statistics and status for the OSPFv2 instance, followed by a brief description of the areas located on the switch.

19.3.5.2 Viewing OSPFv2 on the Interfaces

The **show ip ospf interface** command displays OSPFv2 information for switch interfaces configured for OSPFv2. Different command options allow the display of either all interfaces or a specified interface. The command can also be configured to display complete information or a brief summary.

Example

- This command displays complete OSPFv2 information for VLAN 1.

```
Switch#show ip ospf interface vlan 1
Vlan1 is up, line protocol is up (connected)
  Internet Address 192.168.0.1/24, Area 0.0.0.0
  Process ID 1, Router ID 192.168.103.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router is 192.168.104.2
  Backup Designated router is 192.168.103.1
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 1
  MTU is 1500
Switch#
```

The display indicates the switch is an ABR by displaying a neighbor count, the Designated Router, and Backup Designated Router.

- This command displays a summary of interface information for the switch.

```
Switch#show ip ospf interface brief
Interface    PID   Area            IP Address          Cost  State    Nbrs
Loopback0    1     0.0.0.0         192.168.103.1/24    10    DR       0
Vlan1        1     0.0.0.0         192.168.0.1/24      10    BDR      1
Vlan2        1     0.0.0.2         192.168.2.1/24      10    BDR      1
Vlan3        1     0.0.0.3         192.168.3.1/24      10    DR       0
Switch#
```

Configuration information includes the Process ID (PID), area, IP address, and cost. OSPFv2 operational information includes the Designated Router status and number of neighbors.

19.3.5.3 Viewing the OSPFv2 Database

The **show ip ospf database <link state list>** command displays the LSAs in the LSDB for the specified area. If no area is listed, the command displays the contents of the database for each area on the switch. The database command provides options to display subsets of the LSDB database, a summary of database contents, and the link states that comprise the database.

Examples

- This command displays LSDB contents for area 2.

```
Switch#show ip ospf 1 2 database

            OSPF Router with ID(192.168.103.1) (Process ID 1)

                 Router Link States (Area 0.0.0.2)

Link ID         ADV Router      Age         Seq#       Checksum Link count
192.168.103.1   192.168.103.1   00:29:08    0x80000031 0x001D5F 1
192.168.104.2   192.168.104.2   00:29:09    0x80000066 0x00A49B 1

                 Net Link States (Area 0.0.0.2)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.2.1     192.168.103.1   00:29:08    0x80000001 0x00B89D

                 Summary Net Link States (Area 0.0.0.2)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.0.0     192.168.103.1   00:13:20    0x80000028 0x0008C8
192.168.0.0     192.168.104.2   00:09:16    0x80000054 0x00A2FF
192.168.3.0     192.168.104.2   00:24:16    0x80000004 0x00865F
192.168.3.0     192.168.103.1   00:24:20    0x80000004 0x002FC2
192.168.103.0   192.168.103.1   00:14:20    0x80000028 0x0096D2
192.168.103.0   192.168.104.2   00:13:16    0x80000004 0x00364B
192.168.104.0   192.168.104.2   00:08:16    0x80000055 0x002415
192.168.104.0   192.168.103.1   00:13:20    0x80000028 0x00EF6E
Switch#
```

- This command displays an LSDB content summary for area 2.

```
Switch#show ip ospf 1 2 database database-summary

            OSPF Router with ID(192.168.103.1) (Process ID 1)

Area 0.0.0.2 database summary
  LSA Type      Count
  Router        2
  Network       1
  Summary Net   8
  Summary ASBR  0
  Type-7 Ext    0
  Opaque Area   0
  Subtotal      11

Process 1 database summary
  LSA Type      Count
  Router        2
  Network       1
  Summary Net   8
  Summary ASBR  0
  Type-7 Ext    0
  Opaque Area   0
  Type-5 Ext    0
  Opaque AS     0
  Total         11
Switch#
```

- This command displays the router Link States contained in the area 2 LSDB.

```
Switch#show ip ospf 1 2 database router

            OSPF Router with ID(192.168.103.1) (Process ID 1)

                 Router Link States (Area 0.0.0.2)

  LS age: 00:02:16
  Options: (E DC)
  LS Type: Router Links
  Link State ID: 192.168.103.1
  Advertising Router: 192.168.103.1
  LS Seq Number: 80000032
  Checksum: 0x1B60
  Length: 36
  Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 192.168.2.1
     (Link Data) Router Interface address: 192.168.2.1
      Number of TOS metrics: 0
       TOS 0 Metrics: 10


  LS age: 00:02:12
  Options: (E DC)
  LS Type: Router Links
  Link State ID: 192.168.104.2
  Advertising Router: 192.168.104.2
  LS Seq Number: 80000067
  Checksum: 0xA29C
  Length: 36
  Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 192.168.2.1
     (Link Data) Router Interface address: 192.168.2.2
      Number of TOS metrics: 0
       TOS 0 Metrics: 10
Switch#
```

19.3.5.4 Viewing OSPFv2 Neighbors

The **show ip ospf neighbor** command displays information about the routers that are neighbors to the switch. Command options allow the display of summary or detailed information about the neighbors to all areas and interfaces on the switch. The command also allows for the display of neighbors to individual interfaces or areas. The ***adjacency-changes*** option displays the interface's adjacency changes.

Example

- This command displays the switch's neighbors.

```
Switch#show ip ospf neighbor
Neighbor ID     Pri   State        Dead Time   Address         Interface
192.168.104.2     1   FULL/DR      00:00:35    192.168.0.2     Vlan1
192.168.104.2     8   FULL/BDR     00:00:31    192.168.2.2     Vlan2
Switch#
```

- This command displays details about the neighbors to VLAN 2.

```
Switch#show ip ospf neighbor vlan 2 detail
 Neighbor 192.168.104.2, interface address 192.168.2.2
    In the area 0.0.0.2 via interface Vlan2
    Neighbor priority is 8, State is FULL, 13 state changes
    Adjacency was established 000:01:25:48 ago
    DR is 192.168.2.1 BDR is 192.168.2.2
    Options is E
    Dead timer due in 00:00:34
Switch#
```

- This command displays the adjacency changes to VLAN 2.

```
Switch#show ip ospf neighbor vlan 2 adjacency-changes
[08-04 08:55:32] 192.168.104.2, interface Vlan2 adjacency established
[08-04 09:58:51] 192.168.104.2, interface Vlan2 adjacency dropped:
interface went down
[08-04 09:58:58] 192.168.104.2, interface Vlan2 adjacency established
[08-04 09:59:34] 192.168.104.2, interface Vlan2 adjacency dropped:
interface went down
[08-04 09:59:42] 192.168.104.2, interface Vlan2 adjacency established
[08-04 10:01:40] 192.168.104.2, interface Vlan2 adjacency dropped: nbr did
not list our router ID
[08-04 10:01:46] 192.168.104.2, interface Vlan2 adjacency established
Switch#
```

19.3.5.5 Viewing OSPFv2 Routes

The **show ip routes** command provides an OSPFv2 option.

Examples

- This command displays all of a switch's routes.

```
Switch#show ip route
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP

Gateway of last resort:
S    0.0.0.0/0 [1/0] via 10.255.255.1

C    10.255.255.0/24 is directly connected, Management1
C    192.168.0.0/24 is directly connected, Vlan1
C    192.168.2.0/24 is directly connected, Vlan2
O    192.168.3.0/24 [110/20] via 192.168.0.1
O    192.168.103.0/24 [110/20] via 192.168.0.1
C    192.168.104.0/24 is directly connected, Loopback0
Switch#
```

- This command displays the switch's OSPFv2 routes.

```
Switch#show ip route ospf
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP

O    192.168.3.0/24 [110/20] via 192.168.0.1
O    192.168.103.0/24 [110/20] via 192.168.0.1
Switch#
```

Use the **ping** command to determine the accessibility of a route.

Example

- This command pings an OSPFv2 route.

```
Switch#ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 72(100) bytes of data.
80 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=0.148 ms
80 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=0.132 ms
80 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=0.136 ms
80 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=0.137 ms
80 bytes from 192.168.0.1: icmp_seq=5 ttl=64 time=0.136 ms

--- 192.168.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 7999ms
rtt min/avg/max/mdev = 0.132/0.137/0.148/0.015 ms
Switch#
```

19.4 OSPFv2 Examples

This section describes the commands required to configure three OSPFv2 topologies.

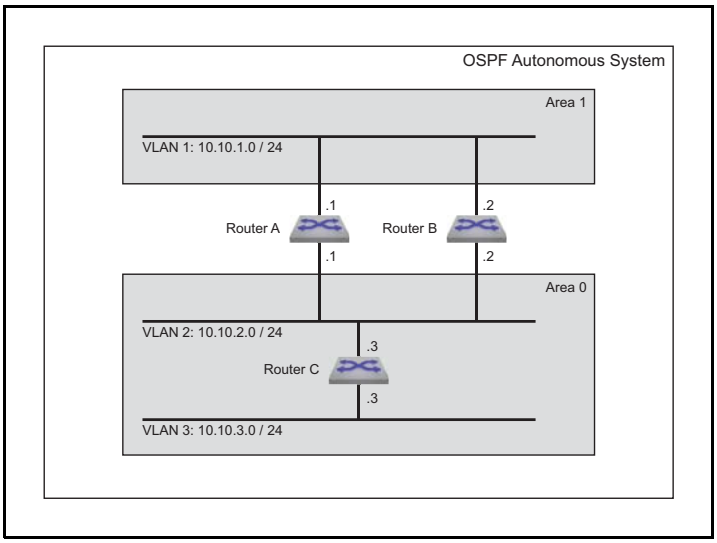
19.4.1 OSPFv2 Example 1

The AS in example 1 contains two areas that are connected through two routers. The backbone area also contains an internal router that connects two subnets.

19.4.1.1 Example 1 Diagram

Figure 19-3 displays the Example 1 topology. Two ABRs connect area 0 and area 1 – Router A and Router B. Router C is an internal router that connects two subnets in area 0.

Figure 19-3 OSPFv2 Example 1



Area 1 Configuration

Area 1 contains one subnet that is accessed by Router A and Router B.

- Router A: The subnet 10.10.1.0/24 is accessed through VLAN 1.
- Router B: The subnet 10.10.1.0/24 is accessed through VLAN 1.
- Each router uses simple authentication, with password abcdefgh.
- Designated Router (DR): Router A.
- Backup Designated Router (BDR): Router B.
- Each router defines an interface cost of 10.
- Router priority is not specified for either router on area 1.

Area 0 ABR Configuration

Area 0 contains one subnet that is accessed by ABRs Router A and Router B.

- Router A: The subnet 10.10.2.0/24 is accessed through VLAN 2.
- Router B: The subnet 10.10.2.0/24 is accessed through VLAN 2.
- Designated Router (DR): Router B.
- Backup Designated Router (BDR): Router A.
- Each router uses simple authentication, with password ijklmnop.
- Each router defines an interface cost of 20.

- Each router defines a retransmit-interval of 10.
- Each router defines a transmit-delay of 2.
- Router priority is specified such that Router B will be elected as the Designated Router.

Area 0 IR Configuration

Area 0 contains one internal router that connects two subnets.

- Router C: The subnet 10.10.2.0/24 is accessed through VLAN 2.
- Router C: The subnet 10.10.3.0/24 is accessed through VLAN 3.
- The subnet 10.10.2.0/24 link is configured as follows:
  — Interface cost of 20.
  — Retransmit-interval of 10.
  — Transmit-delay of 2.
- The subnet 10.10.3.0/24 link is configured as follows:
  — Interface cost of 20.
  — Dead interval of 80 seconds.

19.4.1.2 Example 1 Code

This code configures the OSPFv2 instances on the three switches.

Step 1 Configure the interface addresses.

Step a Router A interfaces:

```
Switch-A(config)#interface vlan 1
Switch-A(config-if-vl1)#ip address 10.10.1.1/24
Switch-A(config-if-vl1)#interface vlan 2
Switch-A(config-if-vl2)#ip address 10.10.2.1/24
```

Step b Router B interfaces:

```
Switch-B(config)#interface vlan 1
Switch-B(config-if-vl1)#ip address 10.10.1.2/24
Switch-B(config-if-vl1)#interface vlan 2
Switch-B(config-if-vl2)#ip address 10.10.2.2/24
```

Step c Router C interfaces:

```
Switch-C(config)#interface vlan 2
Switch-C(config-if-vl2)#ip address 10.10.2.3/24
Switch-C(config-if-vl2)#interface vlan 3
Switch-C(config-if-vl3)#ip address 10.10.3.3/24
```

Step 2 Configure the interface OSPFv2 parameters.

Step a Router A interfaces:

```
Switch-A(config-if-vl2)#interface vlan 1
Switch-A(config-if-vl1)#ip ospf authentication-key abcdefgh
Switch-A(config-if-vl1)#ip ospf cost 10
Switch-A(config-if-vl1)#ip ospf priority 6
Switch-A(config-if-vl1)#interface vlan 2
Switch-A(config-if-vl2)#ip ospf authentication-key ijklmnop
Switch-A(config-if-vl2)#ip ospf cost 20
Switch-A(config-if-vl2)#ip ospf retransmit-interval 10
Switch-A(config-if-vl2)#ip ospf transmit-delay 2
Switch-A(config-if-vl2)#ip ospf priority 4
```

Step b Router B interfaces:

```
Switch-B(config-if-vl2)#interface vlan 1
Switch-B(config-if-vl1)#ip ospf authentication-key abcdefgh
Switch-B(config-if-vl1)#ip ospf cost 10
Switch-B(config-if-vl1)#ip ospf priority 4
Switch-B(config-if-vl1)#interface vlan 2
Switch-B(config-if-vl2)#ip ospf authentication-key ijklmnop
Switch-B(config-if-vl2)#ip ospf cost 20
Switch-B(config-if-vl2)#ip ospf retransmit-interval 10
Switch-B(config-if-vl2)#ip ospf transmit-delay 2
Switch-B(config-if-vl2)#ip ospf priority 6
```

Step c Router C interfaces:

```
Switch-C(config-if-vl3)#interface vlan 2
Switch-C(config-if-vl2)#ip ospf cost 20
Switch-C(config-if-vl2)#ip ospf retransmit-interval 10
Switch-C(config-if-vl2)#ip ospf transmit-delay 2
Switch-C(config-if-vl2)#interface vlan 3
Switch-C(config-if-vl3)#ip ospf cost 20
Switch-C(config-if-vl3)#ip ospf dead-interval 80
```

Step 3 Attach the network segments to the areas.

Step a Router A interfaces:

```
Switch-A(config-if-vl2)#router ospf 1
Switch-A(config-router-ospf)#router-id 169.10.0.1
Switch-A(config-router-ospf)#network 10.10.1.0/24 area 1
Switch-A(config-router-ospf)#network 10.10.2.0/24 area 0
```

Step b Router B interfaces:

```
Switch-B(config-if-vl2)#router ospf 1
Switch-B(config-router-ospf)#router-id 169.10.0.2
Switch-B(config-router-ospf)#network 10.10.1.0/24 area 1
Switch-B(config-router-ospf)#network 10.10.2.0/24 area 0
```

Step c Router C interfaces:

```
Switch-C(config-if-vl3)#router ospf 1
Switch-C(config-router-ospf)#router-id 169.10.0.3
Switch-C(config-router-ospf)#network 10.10.2.0/24 area 0
Switch-C(config-router-ospf)#network 10.10.3.0/24 area 0
```

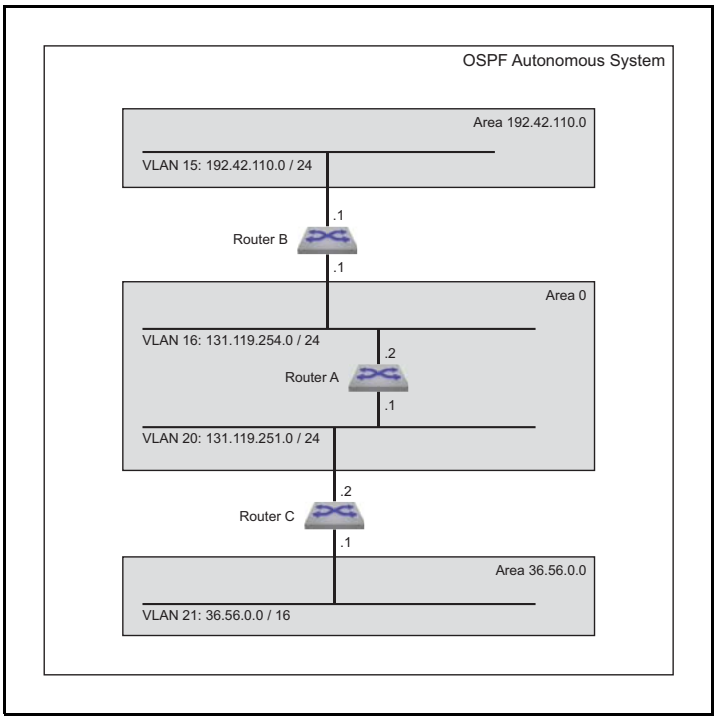
19.4.2 OSPFv2 Example 2

The AS in example 2 contains three areas. Area 0 connects to the other areas through different routers. The backbone area contains an internal router that connects two subnets. Area 0 is normal; the other areas are stub areas.

19.4.2.1 Example 2 Diagram

Figure 19-4 displays the Example 2 topology. One ABR (Router B) connects area 0 and area 192.42.110.0; another ABR (router C) connects area 0 and area 36.56.0.0. Router A is an internal router that connects two subnets in area 0.

Figure 19-4 OSPFv2 Example 2



Area 192.42.110.0 Configuration

Area 192.42.110.0 contains one subnet that is accessed by Router B.

- Router B: The subnet 192.42.110.0 is accessed through VLAN 15.
- Router B uses simple authentication, with password abcdefgh.
- Each router defines a interface cost of 10.

Area 36.56.0.0 Configuration

Area 36.56.0.0 contains one subnet that is accessed by Router C.

- Router C: The subnet 36.56.0.0 is accessed through VLAN 21.
- Router C uses simple authentication, with password ijklmnop.
- Each router defines a interface cost of 20.

Area 0 ABR Configuration

Area 0 contains two subnets. ABR Router B connects one subnet to area 192.42.110.0. ABR Router C connects the other subnet to area 36.56.0.0.

- Router B: The subnet 131.119.254.0/24 is accessed through VLAN 16.
- Router C: The subnet 131.119.251.0/24 is accessed through VLAN 20.
- Designated Router (DR): Router B.
- Backup Designated Router (BDR): Router C.
- Each ABR uses simple authentication, with password ijklmnop
- Each router defines an interface cost of 20.
- Each router defines a retransmit-interval of 10.
- Each router defines a transmit-delay of 2.

Area 0 IR Configuration

Area 0 contains two subnets connected by an internal router.

- Router A: The subnet 131.119.254.0/24 is accessed through VLAN 16.
- Router A: The subnet 131.119.251.0/24 is accessed through VLAN 20.
- The subnet 192.42.110.0 is configured as follows:
  - Interface cost of 10.
- The subnet 36.56.0.0/24 is configured as follows:
  - Interface cost of 20.
  - Retransmit-interval of 10.
  - Transmit-delay of 2.

19.4.2.2 Example 2 Code

Step 1 Configure the interface addresses.

Step a Router A interfaces:

```
Switch-A(config)#interface vlan 16
Switch-A(config-if-vl16)#ip address 131.119.254.2/24
Switch-A(config-if-vl16)#interface vlan 20
Switch-A(config-if-vl20)#ip address 131.119.251.1/24
```

Step b Router B interfaces:

```
Switch-B(config)#interface vlan 15
Switch-B(config-if-vl15)#ip address 192.42.110.1/24
Switch-B(config-if-vl15)#interface vlan 16
Switch-B(config-if-vl16)#ip address 131.119.254.1/24
```

Step c Router C interfaces:

```
Switch-C(config)#interface vlan 20
Switch-C(config-if-vl20)#ip address 131.119.251.2/24
Switch-C(config-if-vl20)#interface vlan 21
Switch-C(config-if-vl21)#ip address 38.56.0.1/24
```

Step 2 Configure the interface OSPFv2 parameters.

Step a Router A interfaces:

```
Switch-A(config-if-vl20)#interface vlan 16
Switch-A(config-if-vl16)#ip ospf cost 10
Switch-A(config-if-vl16)#interface vlan 20
Switch-A(config-if-vl20)#ip ospf cost 20
Switch-A(config-if-vl20)#ip ospf retransmit-interval 10
Switch-A(config-if-vl20)#ip ospf transmit-delay 2
```

Step b Router B interfaces:

```
Switch-B(config-if-vl16)#interface vlan 15
Switch-B(config-if-vl15)#ip ospf authentication-key abcdefgh
Switch-B(config-if-vl15)#ip ospf cost 10
Switch-B(config-if-vl15)#interface vlan 16
Switch-B(config-if-vl16)#ip ospf authentication-key ijklmnop
Switch-B(config-if-vl16)#ip ospf cost 20
Switch-B(config-if-vl16)#ip ospf retransmit-interval 10
Switch-B(config-if-vl16)#ip ospf transmit-delay 2
Switch-B(config-if-vl16)#ip ospf priority 6
```

Step c Router C interfaces:

```
Switch-C(config-if-vl21)#interface vlan 20
Switch-C(config-if-vl20)#ip ospf authentication-key ijklmnop
Switch-C(config-if-vl20)#ip ospf cost 20
Switch-C(config-if-vl20)#ip ospf retransmit-interval 10
Switch-C(config-if-vl20)#ip ospf transmit-delay 2
Switch-C(config-if-vl20)#ip ospf priority 4
Switch-C(config-if-vl20)#interface vlan 21
Switch-C(config-if-vl21)#ip ospf authentication-key ijklmnop
Switch-C(config-if-vl21)#ip ospf cost 20
Switch-C(config-if-vl21)#ip ospf dead-interval 80
```

Step 3 Attach the network segments to the areas.

Step a Router A interfaces:

```
Switch-A(config-if-vl20)#router ospf 1
Switch-A(config-router-ospf)#router-id 174.24.1.1
Switch-A(config-router-ospf)#network 131.119.254.0/24 area 0
Switch-A(config-router-ospf)#network 131.119.251.0/24 area 0
Switch-A(config-router-ospf)#area 0 range 131.119.251.0 0.0.7.255
```

Step b Router B interfaces:

```
Switch-B(config-if-vl16)#router ospf 1
Switch-B(config-router-ospf)#router-id 174.24.1.2
Switch-B(config-router-ospf)#area 192.42.110.0 stub
Switch-B(config-router-ospf)#network 192.42.110.0/24 area 192.42.110.0
Switch-B(config-router-ospf)#network 131.119.254.0/24 area 0
```

Step c Router C interfaces:

```
Switch-C(config-if-vl21)#router ospf 1
Switch-C(config-router-ospf)#router-id 174.24.1.3
Switch-C(config-router-ospf)area 36.56.0.0 stub 0
Switch-C(config-router-ospf)#network 131.119.251.0/24 area 0
Switch-C(config-router-ospf)#network 38.56.0.0/24 area 36.56.0.0
```

19.4.3 OSPFv2 Example 3

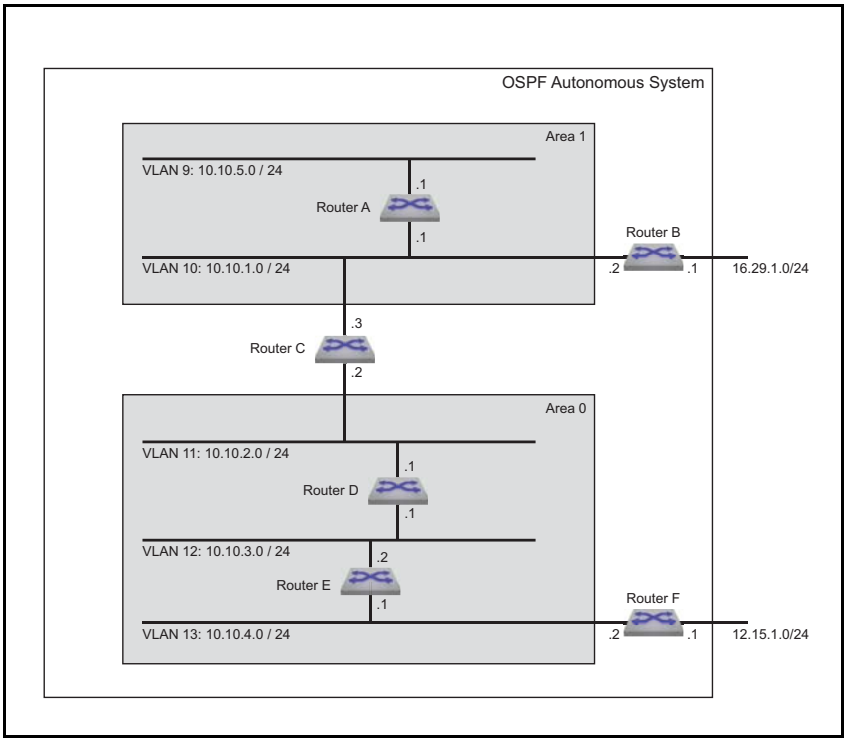
The AS in example 3 contains two areas that connect through one ABR.

- Area 0: Backbone area contains two internal routers that connect three subnets, one ASBR, and one ABR that connects to Area 1.
- Area 1: NSSA contains one internal router, one ASBR, and one ABR that connects to the backbone.

19.4.3.1 Example 3 Diagram

Figure 19-5 displays the Example 3 topology. One ABR connects area 0 and area 1. Router C is an ABR that connects the areas. Router A is an internal router that connects two subnets in area 1. Router D and Router E are internal routers that connect subnets in area 0. Router B and Router F are ASBRs that connect static routes outside the AS to area 1 and area 0, respectively.

Figure 19-5 OSPFv2 Example 3



Area 0 ABR Configuration

ABR Router C connects one area 0 subnet to an area 1 subnet.

- Router C: The subnet 10.10.2.0/24 is accessed through VLAN 11.
- Authentication is not configured on the interfaces.
- All interface OSPFv2 parameters are set to their default values.

Area 0 IR Configuration

Area 0 contains two internal routers, each of which connects two of the three subnets in the area.

- Router D: The subnet 10.10.2.0/24 is accessed through VLAN 11.
- Router D: The subnet 10.10.3.0/24 is accessed through VLAN 12.
- Router E: The subnet 10.10.3.0/24 is accessed through VLAN 12.

- Router E: The subnet 10.10.4.0/24 is accessed through VLAN 13.
- All interface OSPFv2 parameters are set to their default values.

Area 0 ASBR Configuration

ASBR Router F connects one area 0 subnet to an external subnet.

- Router F: The subnet 10.10.4.0/24 is accessed through VLAN 13.
- Router F: The subnet 12.15.1.0/24 is accessed through VLAN 14.
- All interface OSPFv2 parameters are set to their default values.

Area 1 ABR Configuration

ABR Router C connects one area 0 subnet to area 1.

- Router C: The subnet 10.10.1.0/24 is accessed through VLAN 10.
- Authentication is not configured on the interface.
- All interface OSPFv2 parameters are set to their default values.

Area 1 IR Configuration

Area 1 contains one internal router that connects two subnets in the area.

- Router A: The subnet 10.10.1.0/24 is accessed through VLAN 10.
- Router A: The subnet 10.10.5.0/24 is accessed through VLAN 9.
- All interface OSPFv2 parameters are set to their default values.

Area 1 ASBR Configuration

ASBR Router B connects one area 1 subnet to an external subnet.

- Router B: The subnet 10.10.1.0/24 is accessed through VLAN 10.
- Router B: The subnet 16.29.1.0/24 is accessed through VLAN 15.
- All interface OSPFv2 parameters are set to their default values.

19.4.3.2 Example 3 Code

Step 1 Configure the interfaces.

Step a Router A interfaces:

```
Switch-A(config)#interface vlan 10
Switch-A(config-if-vl10)#ip address 10.10.1.1/24
Switch-A(config-if-vl10)#interface vlan 9
Switch-A(config-if-vl11)#ip address 10.10.5.1/24
```

Step b Router B interfaces:

```
Switch-B(config)#interface vlan 10
Switch-B(config-if-vl10)#ip address 10.10.1.2/24
Switch-B(config-if-vl10)#interface vlan 15
Switch-B(config-if-vl18)#ip address 16.29.1.1/24
```

Step c Router C interfaces:

```
Switch-C(config)#interface vlan 10
Switch-C(config-if-vl10)#ip address 10.10.1.3/24
Switch-C(config-if-vl10)#interface vlan 11
Switch-C(config-if-vl11)#ip address 10.10.2.2/24
```

Step d Router D interfaces:

```
Switch-D(config)#interface vlan 11
Switch-D(config-if-vl11)#ip address 10.10.2.1/24
Switch-D(config)#interface vlan 12
Switch-D(config-if-vl12)#ip address 10.10.3.1/24
```

Step e Router E interfaces:

```
Switch-E(config)#interface vlan 12
Switch-E(config-if-vl12)#ip address 10.10.3.2/24
Switch-E(config)#interface vlan 13
Switch-E(config-if-vl13)#ip address 10.10.4.1/24
```

Step f Router F interfaces:

```
Switch-F(config)#interface vlan 13
Switch-F(config-if-vl13)#ip address 10.10.4.2/24
Switch-F(config)#interface vlan 14
Switch-F(config-if-vl14)#ip address 12.15.1.1/24
```

Step 2 Attach the network segments to the areas.

Step a Router A interfaces:

```
Switch-A(config-if-vl10)#router ospf 1
Switch-A(config-router-ospf)#router-id 170.21.0.1
Switch-A(config-router-ospf)#area 1 NSSA
Switch-A(config-router-ospf)#network 10.10.1.0/24 area 1
```

Step b Router B interfaces:

```
Switch-B(config-if-vl10)#router ospf 1
Switch-B(config-router-ospf)#router-id 170.21.0.2
Switch-B(config-router-ospf)#area 1 NSSA
Switch-B(config-router-ospf)#network 10.10.1.0/24 area 1
```

Step c Router C interfaces:

```
Switch-C(config-if-vl11)#router ospf 1
Switch-C(config-router-ospf)#router-id 170.21.0.3
Switch-C(config-router-ospf)#area 1 NSSA
Switch-C(config-router-ospf)#network 10.10.1.0/24 area 1
Switch-C(config-router-ospf)#network 10.10.2.0/24 area 0
```

Step d Router D interfaces:

```
Switch-D(config-if-vl12)#router ospf 1
Switch-D(config-router-ospf)#router-id 170.21.0.4
Switch-D(config-router-ospf)#network 10.10.2.0/24 area 0
Switch-D(config-router-ospf)#network 10.10.3.0/24 area 0
```

Step e Router E interfaces:

```
Switch-E(config-if-vl13)#router ospf 1
Switch-E(config-router-ospf)#router-id 170.21.0.5
Switch-E(config-router-ospf)#network 10.10.3.0/24 area 0
Switch-E(config-router-ospf)#network 10.10.4.0/24 area 0
```

Step f Router F interfaces:

```
Switch-F(config-if-vl14)#router ospf 1
Switch-F(config-router-ospf)#router-id 170.21.0.6
Switch-F(config-router-ospf)#network 10.10.4.0/24 area 0

Switch-F(config-router-ospf)#redistribute static
```

19.5 OSPFv2 Commands

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Mode

- ip ospf name-lookup . . . . . Page 805
- router ospf . . . . . Page 820

Interface Configuration Mode

- ip ospf authentication . . . . . Page 799
- ip ospf authentication-key . . . . . Page 800
- ip ospf cost . . . . . Page 801
- ip ospf dead-interval . . . . . Page 802
- ip ospf hello-interval . . . . . Page 803
- ip ospf message-digest-key . . . . . Page 804
- ip ospf network . . . . . Page 806
- ip ospf priority . . . . . Page 807
- ip ospf retransmit-interval . . . . . Page 808
- ip ospf shutdown . . . . . Page 809
- ip ospf transmit-delay . . . . . Page 810

Router-OSPFv2 Configuration Mode

- area (OSPFv2) . . . . . Page 791
- area default-cost (OSPFv2) . . . . . Page 793
- area filter (OSPFv2) . . . . . Page 794
- area range (OSPFv2) . . . . . Page 795
- auto-cost reference-bandwidth (OSPFv2) . . . . . Page 796
- distance ospf intra-area (OSPFv2) . . . . . Page 797
- exit (OSPFv2) . . . . . Page 798
- log-adjacency-changes (OSPFv2) . . . . . Page 811
- max-lsa (OSPFv2) . . . . . Page 812
- maximum-paths (OSPFv2) . . . . . Page 813
- network area (OSPFv2) . . . . . Page 814
- passive-interface default (OSPFv2) . . . . . Page 815
- passive-interface <interface> (OSPFv2) . . . . . Page 816
- point-to-point routes (OSPFv2) . . . . . Page 817
- redistribute (OSPFv2) . . . . . Page 818
- router-id (OSPFv2) . . . . . Page 819
- shutdown (OSPFv2) . . . . . Page 834
- timers spf (OSPFv2) . . . . . Page 835

Display Commands

- show ip ospf . . . . . Page 821
- show ip ospf border-routers . . . . . Page 822
- show ip ospf database database-summary . . . . . Page 823
- show ip ospf database <link state list> . . . . . Page 824
- show ip ospf database <link-state details> . . . . . Page 825
- show ip ospf interface . . . . . Page 828
- show ip ospf interface brief . . . . . Page 829
- show ip ospf neighbor . . . . . Page 830
- show ip ospf request-list . . . . . Page 832
- show ip ospf retransmission-list . . . . . Page 833

area (OSPFv2)

The **area** command configures the area type of an OSPFv2 area. All routers in an AS must specify the same area type for identically numbered areas.

The switch supports three area types:

- Normal areas: Normal areas accept intra-area, inter-area, and external routes. The backbone (area 0) is a normal area.
- Stub area: Stub areas are areas in which external routes are not advertised. To reach these external routes, a default summary route (0.0.0.0) is inserted into the stub area. Networks without external routes do not require stub areas.
- NSSA (Not So Stubby Area): NSSA ASBRs advertise external LSAs that are part of the area, but do not advertise external LSAs from other areas. An ABR originates the default route, as in stub areas.

Areas are ***normal*** by default; area type configuration is required only for stub and NSSA areas. Area 0 is always a normal area and cannot be configured through this command.

The **no area <type>** and **default area <type>** commands remove the corresponding **area <type>** command from ***running-config***:

- **no/default area nssa translate type7 always** commands remove the **translate type7 always** parameter without changing the area type.
- **no/default area nssa , no/default area stub**, and **no/default area stub no-summary** commands restore the area's type to ***normal***.
- **no/default area** command removes all area commands for the specified area from ***running-config***

Command Mode

Router-OSPF Configuration

Command Syntax

```
area area_id [TYPE]
no area area_id [TYPE]
default area area_id [TYPE]
```

Parameters

- *area_id* area number.

  Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  Running-config stores value in dotted decimal notation.

- ***TYPE*** area type. Values include:

  — **nssa**
  — **nssa translate type7 always** sets p-bit when sending type 7 LSAs
  — **stub**
  — **stub no-summary** Prevents ABRs from sending summary link advertisements into the area.

Guidelines

Type 7 LSA only has area flooding scope. External information is distibuted by translating type 7 LSAs into type 5 LSAs at the NSSA border. The P-bit in the type 7 LSA options field controls the translation of the type 7 LSA; LSAs with the P-bit set are translated.

The no-summary option has the same effect as an **area** *area_id* **filter 0/0** command.

Examples

- This command configures area 45 as a stub area.

```
switch(config)#router ospf 6
switch(config-router-ospf)#area 45 stub
switch(config-router-ospf)#
```

- This command configures area 116.92.148.17 as an NSSA.

```
switch(config-router-ospf)#area 116.92.148.17 NSSA
switch(config-router-ospf)#
```

area default-cost (OSPFv2)

The **area default-cost** command specifies the cost for the default summary routes sent into a specified area. When the **area default-cost** command is not configured for an area, the default-cost of that area is set to 10.

The **no area default-cost** and **default area default-cost** command resets the default-cost value of the specified area to 10 by removing the corresponding **area default-cost** command from ***running-config***. The **no area (OSPFv2)** command removes all area commands for the specified area from ***running-config***, including the **area default-cost** command.

Command Mode

Router-OSPF Configuration

Command Syntax

```
area area_id default-cost def_cost
no area area_id default-cost
default area area_id default-cost
```

Parameters

- *area_id* area number. <**0** to **4294967295**> or <**0.0.0.0** to **255.255.255.255**>
  Running-config stores value in dotted decimal notation.
- *def_cost* cost of the default summary route. Values range from 1 to 65535. Default value is 10.

Example

- This command configures a cost of 15 for default summary routes that an ABR sends into area 23.

```
switch(config)#router ospf 6
switch(config-router-ospf)#area 23 default-cost 15
switch(config-router-ospf)#
```

area filter (OSPFv2)

The **area filter** command prevents an area from receiving Type 3 Summary LSAs from a specified subnet. Type 3 Summary LSAs are sent by ABRs and contain information about one of the areas connected to the ABR. Multiple area filter commands can be configured for an area to specify noncontiguous address spaces.

The **no area filter** and **default area filter** commands remove the specified **area filter** command from ***running-config***. The **no area** command (see **area (OSPFv2)**) removes all area commands for the specified area from ***running-config***, including **area filter** commands.

Command Mode

Router-OSPF Configuration

Command Syntax

```
area area_id filter net_addr
no area area_id filter net_addr
default area area_id filter net_addr
```

Parameters

- *area_id* area number. <**0** to **4294967295**> or <**0.0.0.0** to **255.255.255.255**>
  Running-config stores value in dotted decimal notation.
- *net_addr* network IP address. Entry formats include address-prefix (CIDR) and address-mask.
  Running-config stores value in CIDR notation.

Example

- This command prevents the switch from entering Type 3 LSAs originating from the 10.1.1.0/24 subnet into its area 2 LSDB.

```
switch(config)#router ospf 6
switch(config-router-ospf)#area 2 filter 10.1.1.0/24
switch(config-router-ospf)#
```

area range (OSPFv2)

The **area range** command is used by OSPFv2 area border routers (ABRs) to consolidate or summarize routes, to configure a cost setting for those routes, and to suppress summary route advertisements.

By default, an ABR creates a summary LSA for each route in an area and advertises that LSA to adjacent areas. The **area range** command aggregates routing information on area boundaries, allowing the ABR to use one summary LSA to advertise multiple routes.

The **no area range** and **default area range** commands remove the area-range assignment by deleting the corresponding **area range** command from ***running-config***. The **no area** command (see **area (OSPFv2)**) removes all area commands for the specified area from ***running-config***, including the **area range** command.

Command Mode

Router-OSPF Configuration

Command Syntax

```
area area_id range net_addr [ADVERTISE_SETTING] [COST_SETTING]
no area area_id range net_addr [ADVERTISE_SETTING] [COST_SETTING]
default area area_id range net_addr [ADVERTISE_SETTING] [COST_SETTING]
```

Parameters

- *area_id* area number. <**0** to **4294967295**> or <**0.0.0.0** to **255.255.255.255**>
  Running-config stores value in dotted decimal notation.
- *net_addr* subnet address that includes the summarized routes. Entry formats include address-prefix (CIDR) and address-wildcard mask. ***Running-config*** stores value in CIDR notation.
- ***ADVERTISE_SETTING*** specifies the LSA advertising activity. Values include
  - <no parameter> the switch advertises the address range.
  - **advertise** the switch advertises the address range.
  - **not-advertise** the address range is not advertised to other areas.
- ***COST_SETTING*** specifies the route cost metric for the address range. Values include
  - <no parameter> address range uses the default cost.
  - **cost** *range_cost* assigns cost of *range_cost* to the address range. Value ranges from 1 to 65535.

Examples

- The **network area** commands assign two subnets to an area. The **area range** command summarizes the addresses, which the ABR advertises in a single LSA.

```
switch(config)#router ospf 6
switch(config-router-ospf)#network 10.1.25.80 0.0.0.240 area 5
switch(config-router-ospf)#network 10.1.25.112 0.0.0.240 area 5
switch(config-router-ospf)#area 5 range 10.1.25.64 0.0.0.192
switch(config-router-ospf)#
```

- The **network area** command assigns a subnet to an area, followed by an **area range** command that suppresses the advertisement of that subnet.

```
switch(config-router-ospf)#network 10.12.31.0/24 area 5
switch(config-router-ospf)#area 5 range 10.12.31.0/24 not-advertise
switch(config-router-ospf)#
```

auto-cost reference-bandwidth (OSPFv2)

The **auto-cost reference-bandwidth** command is a factor in the formula that calculates the default OSPFv2 cost for Ethernet interfaces.

OSPFv2-cost = (auto-cost value * 1 Mbps) / interface bandwidth

When auto-cost reference-bandwidth is configured, the switch uses a minimum OSPFv2-cost of one, regardless of the formula parameters. The switch rounds down all non-integer results.

On a 10G Ethernet interface:

- if auto-cost = 100, then OSPFv2-cost = 100 Mbps / 10 Gbps = 0.01, and the default cost is set to 1.
- if auto-cost = 59000, then OSPFv2-cost = 59000 Mbps / 10 Gbps = 5.9, and the default cost is set to 5.

The **ip ospf cost** command configures the OSPFv2 cost for the configuration mode interface and typically reflects the packet transmission overhead for the interface which is inversely proportional to the interface bandwidth. This statement takes precendence over the cost determined by the **auto-cost reference-bandwidth** command.

The **no auto-cost reference-bandwidth** and **default auto-cost reference-bandwidth** command removes the **auto-cost reference-bandwidth** command from ***running-config***. When this parameter is not set, the default cost for Ethernet interfaces is the default **ip ospf cost** value of 10.

Command Mode

Router-OSPF Configuration

Command Syntax

```
auto-cost reference-bandwidth rate
no auto-cost reference-bandwidth
default auto-cost reference-bandwidth
```

Parameters

- *rate* reference bandwidth (megabits/second) Values range from 1 to 4294967 . Default is 100.

Example

To configure a default cost of 20 on 10G Ethernet interfaces:

1. calculate the required auto-cost value:

   auto-cost = (OSPFv2-cost*interface bandwidth)/1 Mbps = (20*10000 Mbps) / 1 Mbps = 200000

2. Configure this value as the auto-cost reference-bandwidth.

```
switch(config)#router ospf 6
switch(config-router-ospf)#auto-cost reference-bandwidht 200000
switch(config-router-ospf)#
```

distance ospf intra-area (OSPFv2)

The **distance ospf intra-area** command specifies the administrative distance for routes contained in a single OSPFv2 area. Administrative distances are used when comparing dynamic routes configured through different protocols. The default administrative distance for intra-area routes is 110.

The **no distance ospf intra-area** and **default distance ospf intra-area** commands remove the **distance ospf intra-area** command from ***running-config***, returning the OSPFv2 distance setting to the default value of 110.

Command Mode

Router-OSPF Configuration

Command Syntax

```
distance ospf intra-area distance
no distance ospf intra-area
default distance ospf intra-area
```

Parameters

- *distance* administrative distance value. Values range from 1 to 255. Default value is 110.

Example

- This command configures a distance of 85 for all OSPFv2 intra-area routes on the switch.

```
switch(config)#router ospf 6
switch(config-router-ospf)#distance ospf intra-area 85
switch(config-router-ospf)#
```

exit (OSPFv2)

In router-ospf configuration mode, the **exit** command places the switch in global configuration mode. Router-ospf configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
exit
```

Example

- This command exits router-ospf configuration mode.

```
switch(config-router-ospf)#exit
switch(config)#
```

ip ospf authentication

The **ip ospf authentication** command enables OSPFv2 authentication for the configuration mode interface. Available authentication methods include simple password and message-digest (MD5). The simple password is configured with the **ip ospf authentication-key** command. The message-digest key is configured with the **ip ospf message-digest-key** command.

The **no ip ospf authentication** and **default ip ospf authentication** commands disable OSPFv2 authentication on the configuration mode interface by removing the corresponding **ip ospf authentication** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf authentication [METHOD]
no ip ospf authentication
default ip ospf authentication
```

Parameters

- *METHOD* OSPFv2 authentication method. Options include:
  - <no parameter> simple password.
  - **message-digest** MD5 authentication.

Examples

- This command enables simple authentication on VLAN 12.

```
switch(config)#interface vlan 12
switch(config-if-vl12)#ip ospf authentication
switch(config-if-vl12)#
```

- This command enables message-digest authentication on VLAN12.

```
switch(config-if-vl12)#ip ospf authentication message-digest
switch(config-if-vl12)#
```

ip ospf authentication-key

The **ip ospf authentication-key** command configures the OSPFv2 authentication password for the configuration mode interface. The plain-text version of the password is a string, up to 8 bytes in length. Interfaces attached to the same area must use the same password to ensure proper communication between neighbors.

OSPFv2 packet headers transmit the password as plain-text, which risks unauthorized password access. ***Running-config*** displays the encrypted version of the password. The encryption scheme is not strong by cryptographic standards; encrypted passwords should be trusted no more than plain-text passwords.

The encryption process uses the interface name as a parameter. Two interfaces with different names cannot use the same encrypted password. However, two interfaces with the same name, but on different switches, can use the same encrypted password.

The **no ip ospf authentication-key** and **default ip ospf authentication-key** commands removes the OSPFv2 authentication password from the configuration mode interface by removing the corresponding **ip ospf authentication-key** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf authentication-key [ENCRYPT_TYPE] key_text
no ip ospf authentication-key
default ip ospf authentication-key
```

Parameters

- ***ENCRYPT_TYPE*** encryption level of the *key_text* parameter. Values include:
  - — <no parameter> the *key_text* is in clear text.
  - — **0** *key_text* is in clear text. Equivalent to <no parameter>.
  - — **7** *key_text* is MD5 encrypted.
- *key_text* the authentication-key password.

Example

- This command specifies a password in clear text.

```
switch(config)#interface vlan 12
switch(config-if-vl12)#ip ospf authentication-key 0 code123
switch(config-if-vl12)#
```

Running-config stores the password as an encrypted string.

ip ospf cost

The **ip ospf cost** command configures the OSPFv2 cost for the configuration mode interface. The OSPFv2 interface cost (or metric) typically reflects the packet transmission overhead for the interface which is inversely proportional to the interface bandwidth. The default cost depends on the interface type:

- Ethernet: determined by the **auto-cost reference-bandwidth (OSPFv2)** command.
- Port channel: 10.
- VLAN: 10.

The **no ip ospf cost** and **default ip ospf cost** commands restore the default OSPFv2 cost for the configuration mode interface by removing the corresponding **ip ospf cost** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf cost interface_cost
no ip ospf cost
default ip ospf cost
```

Parameters

- *interface_cost* cost assigned to the interface. Value ranges from 1 to 65535; default is 10.

Example

- This command configures a cost of 15 for VLAN 2.

```
switch(config)#interface vlan 2
switch(config-if-Vl2)#ip ospf cost 15
switch(config-if-Vl2)#
```

ip ospf dead-interval

The **ip ospf dead-interval** command configures the dead interval for the configuration mode interface.

The dead interval specifies the period that an interface waits for an OSPFv2 packet from a neighbor before it disables the adjacency under the assumption that the neighbor is down. The dead interval should be configured identically on all OSPFv2 neighbors and be longer than the hello interval of any neighbor.

The **no ip ospf dead-interval** and **default ip ospf dead-interval** commands restore the default dead interval of 40 seconds on the configuration mode interface by removing the corresponding **ip ospf dead-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf dead-interval time
no ip ospf dead-interval
default ip ospf dead-interval
```

Parameters

- *time* dead interval (seconds). Value ranges from 1 to 8192; default is 40.

Example

- This command configures a dead interval of 120 seconds for VLAN 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip ospf dead-interval 120
switch(config-if-Vl4)#
```

ip ospf hello-interval

The **ip ospf hello-interval** command configures the OSPFv2 hello interval for the configuration mode interface. The hello interval defines the period between the transmission of consecutive hello packets.

Each OSPFv2 neighbor should specify the same hello interval, which should not be longer than any neighbor's dead interval.

The **no ip ospf hello-interval** and **default ip ospf hello-interval** commands restore the default hello interval of 10 seconds on the configuration mode interface by removing the **ip ospf hello-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf hello-interval time
no ip ospf hello-interval
default ip ospf hello-interval
```

Parameters

- *time* hello interval (seconds). Values range from 1 to 8192; default is 10.

Example

- This command configures a hello interval of 30 seconds for VLAN 2.

```
switch(config)#interface vlan 2
switch(config-if-Vl2)#ip ospf hello-interval 30
switch(config-if-Vl2)#
```

ip ospf message-digest-key

The **ip ospf message-digest-key** command configures a message digest authentication key for the configuration mode interface.

Each interface is configured with a key (password) and key ID pair. When transmitting a packet, the interface generates a message digest string, using the MD5 algorithm, based on the OSPFv2 packet, key, and key ID, then appends that string to the packet.

Message digest authentication supports uninterrupted transmissions during key changes by allowing each interface to have two MD5 keys, each with a different key ID. When a new key is configured on an interface, the router transmits OSPFv2 packets for both keys. The router stops sending duplicate packets when it detects that all of its neighbors have the same key.

The **no ip ospf message-digest-key** and **default ip ospf message-digest-key** commands remove the message digest authentication key for the specified key ID on the configuration mode interface by deleting the corresponding **ip ospf message-digest-key** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf message-digest-key key_id md5 ENCRYPT_TYPE key_text
no ip ospf message-digest-key key_id
default ip ospf message-digest-key key_id
```

Parameters

- *key_id* key ID number. Value ranges from 1 to 255.
- ***ENCRYPT_TYPE*** encryption level of the *key_text* parameters. Values include:
  - <no parameter> *key_text* is unencrypted clear text.
  - **0** *key_text* is unencrypted clear text. Equivalent to <no parameter>.
  - **7** *key_text* must be entered as an MD5 encrypted string.
- *key_text* message digest key (password).

Example

- This command configures ***code123*** as the MD5 key with a corresponding key ID of 23.

```
switch(config)#interface vlan 12
switch(config-if-vl12)#ip ospf message-digest-key 23 md5 0 code123
switch(config-if-vl12)#
```

Running-config stores the password as an encrypted string.

ip ospf name-lookup

The **ip ospf name-lookup** command causes the switch to display DNS names in place of numeric OSPFv2 router IDs in all subsequent OSPFv2 show commands, including:

- **show ip ospf**
- **show ip ospf border-routers**
- **show ip ospf database <link state list>**
- **show ip ospf database database-summary**
- **show ip ospf database <link-state details>**
- **show ip ospf interface**
- **show ip ospf neighbor**
- **show ip ospf request-list**
- **show ip ospf retransmission-list**

Although this command makes it easier to identify a router, the switch relies on a configured DNS server to respond to reverse DNS queries, which may be slower than displaying numeric router IDs.

The **no ip ospf name-lookup** and **default ip ospf name-lookup** commands remove the **ip ospf name-lookup** command from ***running-config***, restoring the default behavior of displaying OSPFv2 router IDs by their numeric value.

Command Mode

Global Configuration

Command Syntax

```
ip ospf name-lookup
no ip ospf name-lookup
default ip ospf name-lookup
```

Example

- This command programs the switch to display OSPFv2 router IDs by the corresponding DNS name in subsequent show commands.

```
switch(config)#ip ospf lookup
switch(config)#
```

ip ospf network

The **ip ospf network** command sets the configuration mode interface as a point-to-point link. By default, interfaces are configured as broadcast links.

The **no ip ospf network** and **default ip ospf network** commands set the configuration mode interface as a broadcast link by removing the corresponding **ip ospf network** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf network point-to-point
no ip ospf network
default ip ospf network
```

Examples

- These commands configure Ethernet interface 10 as a point-to-point link.

```
switch(config)#interface ethernet 10
switch(config-if-Et10)#ip ospf network point-to-point
switch(config-if-Et10)#
```

- This command restores Ethernet interface 10 as a broadcast link.

```
switch(config-if-Et10)#no ip ospf network
switch(config-if-Et10)#
```

ip ospf priority

The **ip ospf priority** command configures OSPFv2 router priority for the configuration mode interface.

Router priority determines preference during designated router (DR) and backup designated router (BDR) elections. Routers with higher priority numbers have preference over other routers. The default priority is 1. Routers with a priority of zero cannot be elected as a DR or BDR.

The **no ip ospf priority** and **default ip ospf priority** commands restore the default priority (1) on the configuration mode interface by removing the corresponding **ip ospf priority** command from *running-config*

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf priority priority_level
no ip ospf priority
default ip ospf priority
```

Parameters

- *priority_level* priority level. Settings range from 0 to 255.

Examples

- This command configures a router priority of 15 for VLAN 8.

```
switch(config)#interface vlan 8
switch(config-if-Vl8)#ip ospf priority 15
switch(config-if-Vl8)#
```

- This command restores the router priority of 1 for VLAN 7.

```
switch(config)#interface vlan 7
switch(config-if-Vl7)#no ip ospf priority
switch(config-if-Vl7)#
```

ip ospf retransmit-interval

The **ip ospf retransmit-interval** command configures the link state advertisement (LSA) retransmission interval for the configuration mode interface.

Routers that send LSAs to an adjacent router expect to receive an acknowledgment from that neighbor. Routers that do not receive an acknowledgment will retransmit the LSA. The retransmission interval specifies the period between these transmissions.

The **no ip ospf retransmit-interval** and **default ip ospf retransmit-interval** commands restore the default retransmission interval of 5 seconds on the configuration mode interface by removing the corresponding **ip ospf retransmit-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf retransmit-interval period
no ip ospf retransmit-interval
default ip ospf retransmit-interval
```

Parameters

- *period* retransmission interval (seconds). Value ranges from 1 to 8192; default is 5.

Example

- This command configures a retransmission interval of 15 seconds for VLAN 3.

```
switch(config)#interface vlan 3
switch(config-if-Vl3)#ip ospf retransmit-interval 15
switch(config-if-Vl3)#
```

ip ospf shutdown

The **ip ospf shutdown** command disables OSPFv2 on the configuration mode interface without disrupting the OSPFv2 configuration. When OSPFv2 is enabled on the switch, the it is also enabled by default on all interfaces.

Neighbor routers are notified of the shutdown and all traffic that has another path through the network will be directed to an alternate path.

The OSPFv2 instance is disabled on the entire switch with the **shutdown (OSPFv2)** command.

The **no ip ospf shutdown** and **default ip ospf shutdown** commands enable OSPFv2 on the configuration mode interface by removing the corresponding **ip ospf shutdown** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf shutdown
no ip ospf shutdown
default ip ospf shutdown
```

Examples

- This command shuts down OSPFv2 activity on VLAN 5.

```
switch(config)#interface vlan 5
switch(config-if-Vl5)#ip ospf shutdown
switch(config-if-Vl5)#
```

- This command resumes OSPFv2 activity on VLAN 5.

```
switch(config-if-Vl5)#no ip ospf shutdown
switch(config-if-Vl5)#
```

ip ospf transmit-delay

The **ip ospf transmit-delay** command configures the transmission delay for OSPFv2 packets over the configuration mode interface.

The transmission delay is an estimate of the time that an interface requires to transmit a link-state update packet. OSPFv2 adds this delay to the age of outbound packets to more accurately reflect the age of the LSA when received by a neighbor.

The **no ip ospf transmit-delay** and **default ip ospf transmit-delay** commands restore the default transmission delay (one second) on the configuration mode interface by removing the corresponding **ip ospf transmit-delay** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip ospf transmit-delay trans
no ip ospf transmit-delay
default ip ospf transmit-delay
```

Parameters

- *trans* LSA transmission delay (seconds). Value ranges from 1 to 8192; default is 1.

Example

- This command configures a transmission delay of 5 seconds for VLAN 6.

```
switch(config)#interface vlan 6
switch(config-if-Vl6)#ip ospf transmit-delay 5
switch(config-if-Vl6)#
```

log-adjacency-changes (OSPFv2)

The **log-adjacency-changes** command configures the switch to send syslog messages either when it detects OSPFv2 link state changes or when it detects that a neighbor has gone up or down. Log message sending is enabled by default. Valid options include:

- **log-adjacency-changes:** switch sends syslog messsages when a neighbor goes up or down (default).
- **log-adjacency-changes detail**: switch sends syslog messages on an OSPFv2 link state change.
- **no log-adjacency-changes** disables link state change syslog reporting.

The default option is active when ***running-config*** does not contain any form of the command. Entering the command in any form replaces the previous command state in ***running-config***. The **default log-adjacency-changes** command restores the default state by removing the **log-adjacency-changes detail** or **no log-adjacency-changes** statement from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
log-adjacency-changes
log-adjacency-changes detail
no log-adjacency-changes
default log-adjacency-changes
```

Examples

- This command configures the switch to send a syslog message when a neighbor goes up or down.

```
switch(config)#router ospf 6
switch(config-router-ospf)#log-adjacency-changes
switch(config-router-ospf)#
```

After entering the command, **show active** does not display a **log-adjacency-changes** statement.

```
switch(config-router-ospf)#show active
router ospf 1
switch(config-router-ospf)#
```

- This command configures the switch to send a syslog message when it detects any link state change.

```
switch(config-router-ospf)#log-adjacency-changes detail
switch(config-router-ospf)#
```

After entering the command, **show active** displays a **log-adjacency-changes detail** command.

```
switch(config-router-ospf)#show active
router ospf 1
   log-adjacency-changes detail
switch(config-router-ospf)#
```

max-lsa (OSPFv2)

The **max-lsa** command specifies the number of LSAs allowed in the LSDB and configures switch actions when the limit is approached or exceeded. Setting the LSA limit to zero removes the LSDB size restriction and disables LSA overload actions. Actions triggered by LSDB overload conditions include:

- **Warning**: LSDB size exceeds the warning threshold – an OSPF MAXLSAWARNING is logged.
- **Temporary shutdown**: LSDB size exceeds specified maximum – OSPFv2 is disabled for a specified period during which it does not accept or acknowledge new LSAs.
- **Permanent shutdown**: A specified number of temporary shutdowns during a given period permanently disables OSPFv2; a **router OSPF** command is required to enable OSPFv2.

The **no max-lsa** and **default max-lsa** commands restore all LSA overload parameters to their default settings by placing the **max-lsa 12000** statement in ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
max-lsa lsa_num [WARNING] [IGNORE_TIME] [IGNORE_COUNT] [RESET]
no max-lsa
default max-lsa
```

Parameters

- *lsa_num* maximum number of LSAs. Value ranges from 0 to 100,000.
  - **0** disables LSA overload protection by specifying an unlimited number of LSAs.
  - *1-100000* specifies the LSA limit; default value is 12,000.
- ***WARNING*** warning threshold, as a percentage of the maximum number of LSAs (% of *lsa_num*).
  - <no parameter> warning threshold set to default of 75%.
  - *percent* percentage. *percent* ranges from 25 to 99.
- ***IGNORE_TIME*** temporary shutdown period (minutes). Options include:
  - <no parameter> temporary shutdown set to default value of 5 minutes.
  - **ignore-time** *period* temporary shutdown set to *period*. Value ranges from 1 to 60.
- ***IGNORE_COUNT*** number of temporary shutdowns required to trigger a permanent shutdown.
  - <no parameter> temporary shutdown counter set to default value of 5.
  - **ignore-count** *episodes* temporary shutdown counter set to *episodes;* ranges from 1 to 20.
- ***RESET*** period of not exceeding LSA limit required to reset temporary shutdown counter to zero.
  - <no parameter> reset timer set to default value of 5 minutes
  - **reset-time** *r_period* reset timer set to *r_period* (minutes). *r_period* ranges from 1 to 60.

Example

- This command defines an LSA limit of 20,000 and configures these actions.
  - Logs an OSPF MAXLSAWARNING message after receiving 8,000 LSAs (40% of 20,000).
  - Disables OSPFv2 for 10 minutes after it receives 20,000 LSA packets.
  - Permanently disables OSPFv2 after four temporary OSPFv2 shutdowns.
  - Resets the shutdown counter to zero if the LSA limit is not exceeded for 20 minutes.

```
s(config-router-ospf)#max-lsa 20000 40 ignore-time 10 ignore-count 4 reset-time 20
s(config-router-ospf)#
```

maximum-paths (OSPFv2)

The **maximum-paths** command controls the maximum number of parallel routes that OSPFv2 supports on the switch. The default maximum is 16 paths.

The **no maximum-paths** and **default maximum-paths** commands restore the maximum number of parallel routes that OSPFv2 supports on the switch to the default value of 16 by placing the **maximum-paths 16** statement in ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
maximum-paths paths
no maximum-paths
default maximum-paths
```

Parameters

- *paths* maximum number of parallel routes. Values range from 1 to 16.

Example

- This command configures the maximum number of OSPFv2 parallel paths to 12.

```
switch(config)#router ospf 6
switch(config-router-ospf)#maximum-paths 12
switch(config-router-ospf)#
```

network area (OSPFv2)

The **network area** command assigns the specified IPv4 subnet to an OSPFv2 area. ***Running-config*** zeroes the host portion of the address; for example, 1.2.3.4/24 is saved as 1.2.3.0/24.

The **no network area** and **default network area** commands delete the specified network area assignment by removing the corresponding **network area** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
network ipv4_subnet area area_id
no network ipv4_subnet area area_id
default network ipv4_subnet area area_id
```

Parameters

- *ipv4_subnet* IPv4 subnet. Entry formats include address-prefix (CIDR) or address-wildcard mask. ***Running-config*** stores value in CIDR notation.
- *area_id* area number. <**0** to **4294967295**> or <**0.0.0.0** to **255.255.255.255**> ***Running-config*** stores value in dotted decimal notation.

Example

- These equivalent commands each assign the subnet 10.1.10.0/24 to area 0.

```
switch(config-router-ospf)#network 10.1.10.0 0.0.0.255 area 0
switch(config-router-ospf)#

switch(config-router-ospf)#network 10.1.10.0/24 area 0
switch(config-router-ospf)#
```

passive-interface default (OSPFv2)

The **passive-interface default** command configures the OSPFv2 instance such that all interfaces are OSPFv2 passive by default. The router neither sends OSPFv2 packets, nor processes OSPFv2 packets received on passive interfaces. The router advertises the passive interface as part of the router LSA.

The **passive-interface <interface> (OSPFv2)** configures the OSPFv2 active-passive status for a specific interface:

- When ***passive-interface default*** is not set, all interfaces are OSPFv2 active by default and passive interfaces are denoted by **passive-interface <interface>** statements in ***running-config***.
- When ***passive-interface default*** is set, all interfaces are OSPFv2 passive by default and active interfaces are denoted by **no passive-interface <interface>** statements in ***running-config***.

The **no passive-interface** and **default passive-interface** commands sets the interface setting such that all interfaces are OSPFv2 active by default by removing the **passive-interface default** statement from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
passive-interface default
no passive-interface default
default passive-interface default
```

Examples

- This command configures the default interface setting as OSPFv2 passive. This command also removes all **passive-interface <interface>** statements from ***running-config***.

```
switch(config)#router ospf 6
switch(config-router-ospf)#passive-interface default
switch(config-router-ospf)#
```

- This command configures the default interface setting as OSPFv2 active. This command also removes all **no passive-interface <interface>** statements from ***running-config***.

```
switch(config-router-ospf)#no passive-interface default
switch(config-router-ospf)#
```

passive-interface <interface> (OSPFv2)

The **passive-interface** command disables OSPFv2 processing on an interface range. The router neither sends OSPFv2 packets, nor processes OSPFv2 packets received on passive interfaces. The router advertises the passive interface as part of the router LSA.

The default OSPFv2 interface activity setting is configured by the **passive-interface default (OSPFv2)** command:

- When ***passive-interface default*** is not set, all interfaces are OSPFv2 active by default and passive interfaces are denoted by **passive-interface <interface>** statements in ***running-config***.
- When ***passive-interface default*** is set, all interfaces are OSPFv2 passive by default and active interfaces are denoted by **no passive-interface <interface>** statements in ***running-config***.

The **no passive-interface** command enables OSPFv2 processing on the specified interface range. The **default passive-interface** command sets the interface to the default interface activity setting by removing the corresponding **passive-interface** or **no passive-interface** statement from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
passive-interface INTERFACE_NAME
no passive-interface INTERFACE_NAME
default passive-interface INTERFACE_NAME
```

Parameters

- *INTERFACE_NAME* interface to be configured. Options include:
  - — **ethernet** *e_range* Ethernet interface list.
  - — **port-channel** *p_range* Channel group interface list.
  - — **vlan** *v_range* VLAN interface list.

  Valid *e_range*, *p_range*, and *v_range* formats include number, range, or comma-delimited list of numbers and ranges.

Examples

- These commands configure Ethernet interfaces 2 through 5 as passive interfaces.

```
switch(config)#router ospf 6
switch(config-router-ospf)#passive-interface ethernet 2-5
switch(config-router-ospf)#
```

- This command configures VLAN interfaces 50-54, 61, 68, and 102-120 as passive interfaces.

```
switch(config-router-ospf)#passive-interface vlan 50-54,61,68,102-120
switch(config-router-ospf)#
```

- This command configures VLAN 2 as an active interface.

```
switch(config-router-ospf)#no passive-interface vlan 2
switch(config-router-ospf)#
```

point-to-point routes (OSPFv2)

When OSPFv2 is enabled, the switch maintains a local routing information base (RIB) to store routes to destinations that it learns from its neighbors. After each calculation, OSPFv2 attempts to install the least-cost routes. By default, the RIB includes point-to-point links that are in the network. The **no point-to-point routes** command optimizes the RIB table by not installing point-to-point links.

The **point-to-point routes** and **default point-to-point routes** commands program the switch to include point-to-point links in its RIB by removing the **no point-to-point routes** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
point-to-point routes
no point-to-point routes
default point-to-point routes
```

Examples

- This command configures the switch to optimize the local RIB by not including point-to-point routes.

```
switch(config)#router ospf 6
switch(config-router-ospf)#no point-to-point routes
switch(config-router-ospf)#
```

- This command configures the switch to include point-to-point routes.

```
switch(config-router-ospf)#point-to-point routes
switch(config-router-ospf)#
```

redistribute (OSPFv2)

The **redistribute** command enables the advertising of all specified routes on the switch into the OSPFv2 domain as external routes. Each command enables the redistribution of one route type. ***Running-config*** allows multiple **redistribute** commands, one for each type of route to be redistributed into the OSPFv2 domain. Individual routes are not configurable for redistribution.

The **no redistribute** and **default redistribute** commands remove the corresponding **redistribute** command from ***running-config***, disabling route redistribution for the specified route type.

Command Mode

Router-OSPF Configuration

Command Syntax

```
redistribute ROUTE_TYPE [ROUTE_MAP]
no redistribute ROUTE_TYPE
default redistribute ROUTE_TYPE
```

Parameters

- ***ROUTE_TYPE*** source from which routes are redistributed. Options include:
  - **aggregate** BGP aggregate routes.
  - **connected** routes that are established when IPv4 is enabled on an interface.
  - **BGP** routes from a BGP domain.
  - **RIP** routes from a RIP domain.
  - **static** IP static routes.
- ***ROUTE_MAP*** route map that determines the routes that are redistributed. Options include:
  - <no parameter > all routes are redistributed.
  - **route-map** *map_name* only routes in the specified route map are redistributed.

Examples

- The **redistribute static** command starts the advertising of static routes as OSPFv2 external routes.

```
switch(config)#router ospf 6
switch(config-router-ospf)#redistribute static
switch(config-router-ospf)#
```

- The **no redistribute bgp** command stops the advertising of BGP routes as OSPFv2 external routes.

```
switch(config-router-ospf)#no redistribute bgp
switch(config-router-ospf)#
```

router-id (OSPFv2)

The **router-id** command configures the router ID for an OSPFv2 instance. The router ID is a 32-bit number, expressed in dotted decimal notation, similar to an IP address. This number uniquely identifies the router within an Autonomous System. Status commands use the router ID to identify the switch.

The switch sets the router ID to the first available alternative in the following list:

1. The **router-id** command.
2. The loopback IP address, if a loopback interface is configured on the switch.
3. The highest IP address present on the router.

The **no router-id** and **default router-id** commands remove the router ID command from ***running-config***; the switch uses the loopback or highest address as the router ID.

Command Mode

Router-OSPF Configuration

Command Syntax

```
router-id identifier
no router-id [identifier]
default router-id [identifier]
```

Parameters

- *identifier* router ID (dotted decimal notation). Value ranges from 0.0.0.0 to 255.255.255.255.

Example

- This command assigns 15.5.4.2 as the router ID for the OSPFv2 instance.

```
switch(config)#router ospf 6
switch(config-router-ospf)#router-id 15.5.4.2
switch(config-router-ospf)#
```

router ospf

The **router ospf** command places the switch in Router OSPF configuration mode and, if the switch does not contain an OSPFv2 instance, instantiates OSPFv2 and provides a process ID for the new instance. The **exit (OSPFv2)** command returns the switch to global configuration mode.

The switch supports one OSPFv2 instance, identified by its process ID. When an instance exists, this command must specify its process ID. Attempts to create additional instances will generate errors. Process IDs are local to the switch and have no effect on instances in the same AS on different routers.

The **show ip ospf** command displays the process ID of the OSPFv2 instance configured on the switch.

The **no router ospf** and **default router ospf** commands deletes the OSPFv2 instance.

These commands are available in router-ospf configuration modes

- **area (OSPFv2)**
- **area default-cost (OSPFv2)**
- **area filter (OSPFv2)**
- **area range (OSPFv2)**
- **auto-cost reference-bandwidth (OSPFv2)**
- **distance ospf intra-area (OSPFv2)**
- **exit (OSPFv2)**
- **log-adjacency-changes (OSPFv2)**
- **max-lsa (OSPFv2)**
- **maximum-paths (OSPFv2)**
- **network area (OSPFv2)**
- **passive-interface default (OSPFv2)**
- **passive-interface <interface> (OSPFv2)**
- **point-to-point routes (OSPFv2)**
- **redistribute (OSPFv2)**
- **router-id (OSPFv2)**
- **shutdown (OSPFv2)**
- **timers spf (OSPFv2)**

Command Mode

Global Configuration

Command Syntax

```
router ospf process_id
no router ospf process_id
default router ospf process_id
```

Parameters

- *process_id* OSPFv2 process ID. Values range from 1 to 65535.

Examples

- This command creates an OSPFv2 instance with process ID 145.

```
switch(config)#router ospf 145
switch(config-router-ospf)#
```

- This command deletes the OSPFv2 instance.

```
switch(config)#no router ospf 145
switch(config)#
```

show ip ospf

The **show ip ospf** command displays general information about switch OSPFv2 routing processes.

Command Mode

EXEC

Command Syntax

show ip ospf [*process_id*]

Parameters

- *process_id* OSPFv2 process ID. Values include:
  - — <no parameter> Command returns data for all OSPFv2 instances.
  - — 1 to 65535 Command returns data for specified OSPFv2 instance.

Example

- This command displays configuration parameters, operational statistics, status of the OSPFv2 instance, and a brief description of the areas on the switch.

```
switch>show ip ospf
Routing Process "ospf 1" with ID 192.168.103.1
 Supports opaque LSA
 Maximum number of LSA allowed 12000
  Threshold for warning message 75%
  Ignore-time 5 minutes, reset-time 5 minutes
  Ignore-count allowed 5, current 0
 It is an area border router
 Hold time between two consecutive SPFs 5000 msecs
 SPF algorithm last executed 00:00:09 ago
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of LSA 27.
 Number of areas in this router is 3. 3 normal 0 stub 0 nssa
    Area BACKBONE(0.0.0.0)
        Number of interfaces in this area is 2
        It is a normal area
        Area has no authentication
        SPF algorithm executed 153 times
        Number of LSA 8. Checksum Sum 0x03e13a
        Number of opaque link LSA 0. Checksum Sum 0x000000
    Area 0.0.0.2
        Number of interfaces in this area is 1
        It is a normal area
        Area has no authentication
        SPF algorithm executed 153 times
        Number of LSA 11. Checksum Sum 0x054e57
        Number of opaque link LSA 0. Checksum Sum 0x000000
    Area 0.0.0.3
        Number of interfaces in this area is 1
        It is a normal area
        Area has no authentication
        SPF algorithm executed 5 times
        Number of LSA 6. Checksum Sum 0x02a401
        Number of opaque link LSA 0. Checksum Sum 0x000000
```

show ip ospf border-routers

The **show ip ospf border-routers** command displays the internal OSPFv2 routing table entries to area border routers (ABRs) and autonomous system boundary routers (ASBRs) for each OSPFv2 area.

Command Mode

EXEC

Command Syntax

```
show ip ospf border-routers
```

Example

- This command displays the ABRs and ASBRs configured in the switch.

```
switch>show ip ospf border-routers
OSPF Process 172.17.0.42

Router ID       Area            Type
172.17.0.1      0.0.0.0         ASBR
switch>
```

show ip ospf database database-summary

The **show ip ospf database database-summary** command displays the number of link state advertisements (LSAs), by type and total, in the OSPFv2 database. The switch can return data about a single area or for all areas on the switch.

Command Mode

EXEC

Command Syntax

show ip ospf [***AREA***] **database database-summary**

Parameters

- ***AREA*** areas for which command displays data. Specifying an individual area requires entering the process ID where the area is located. Options include:
  - — <no parameter> data is displayed for all areas.
  - — *process_id* data is displayed for all areas in specified process ID.
  - — *process_id area_id* data is displayed for specified area.

  process_id input range: <1 to 65535>
  area_id input range: <0 to 4294967295> or <0.0.0.0 to 255.255.255.255>

Example

- This command displays the LSDB content summary for area 0.

```
switch>show ip ospf 1 0 database database-summary

  LSA Type      Count
  Router        18
  Network       21
  Summary Net   59
  Summary ASBR  4
  Type-7 Ext    0
  Opaque Area   0
  Type-5 Ext    4238
  Opaque AS     0
  Total         4340

switch>
```

show ip ospf database <link state list>

The **show ip ospf database <link state list>** command displays the OSPFv2 link state advertisements (LSAs) that originate on a specified switch or router. The command displays data about a single area or for all areas on the switch.

Command Mode

EXEC

Command Syntax

show ip ospf [***AREA***] **database** [***ROUTER***]

Parameters

- ***AREA*** areas for which command displays data. Specifying an individual area requires entering the process ID where the area is located. Options include:
  - <no parameter> command returns data for all areas.
  - *process_id* command returns data for all areas in the specified process ID.
  - *process_id area_id* command returns data for specified area in the specified process ID.

  process_id value ranges from 1 to 65535.
  area_id is entered in decimal or dotted decimal notation.

- ***ROUTER*** router or switch for which the command provides data. Options include:
  - <no parameter> all routers in the specified areas.
  - **adv-router** [*a.b.c.d*] an external router. Specifies local switch if an IP address is not included.
  - **self-originate** local switch. Equivalent to ***adv-router*** option without an IP address.

Example

- This command displays OSPFv2 LSAs that originate at the router with a router ID of 172.26.0.31.

```
switch>show ip ospf database adv-router 172.26.0.31

            OSPF Router with ID(172.26.0.23) (Process ID 1)

172.26.0.31     172.26.0.31     918         0x80002b4a   0x1315   3

                 Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#         Checksum
172.24.238.238  172.26.0.31     678         0x800003d2   0x8acf   0
172.24.238.244  172.26.0.31     678         0x800003d2   0x4e06   0
172.24.238.248  172.26.0.31     678         0x800003d2   0x262a   0
172.24.238.240  172.26.0.31     678         0x800003d2   0x76e1   0
172.24.238.224  172.26.0.31     678         0x800003d2   0x1751   0
<-------OUTPUT OMITTED FROM EXAMPLE-------->
172.24.238.126  172.26.0.31     678         0x800003d2   0xeedb   0
172.24.238.62   172.26.0.31     678         0x800003d2   0x7199   0
172.24.238.254  172.26.0.31     678         0x800003d2   0xe960   0

  Type 11 Opaque LSDB

Type         Link ID      ADV Router   Age       Seq# Checksum
switch>
```

show ip ospf database <link-state details>

The **show ip ospf database <link-state details>** command displays details of the specified link state advertisements (LSAs). The switch can return link state data about a single area or for all areas on the switch.

Command Mode

EXEC

Command Syntax

```
show ip ospf [AREA] database LINKSTATE_TYPE linkstate_id [ROUTER]
```

Parameters

- ***AREA*** areas for which command displays data. Specifying an individual area requires entering the process ID where the area is located. Options include:
  - <no parameter> data is displayed for all areas.
  - *process_id* data is displayed for all areas in specified process ID.
  - *process_id area_id* data is displayed for specified area.

  process_id input range: <1 to 65535>
  area_id input range: <0 to 4294967295> or <0.0.0.0 to 255.255.255.255>

- ***LINKSTATE_TYPE*** link state types. Parameter options include:
  - **detail** Displays all link states.
  - **router** Displays the Type 1 (Router) link states.
  - **network** Displays the Type 2 (Network) link states.
  - **summary** Displays the Type 3 (Summary) link states.
  - **asbr-summary** Displays the Type 4 (ASBR-Summary) link states.
  - **external** Displays the Type 5 (External) link states.
  - **nssa-external** Displays the Type 7 (NSSA-External) link states.
  - **opaque-link** Displays the Type 9 (Link-Local Opaque) link states.
  - **opaque-area** Displays the Type 10 (Area-Local Opaque) link states.
  - **opaque-as** Displays the Type 11 (AS Opaque) link states.
- *linkstate_id* Network segment described by the LSA (dotted decimal notation).

  Value depends on the LSA type.
  - When the LSA describes a network, the *linkstate-id* argument is one of the following:

    The network IP address, as in Type 3 summary link advertisements and in autonomous system external link advertisements.

    A derived address obtained from the link state ID. Masking a network links the advertisement link state ID with the network subnet mask yielding the network IP address.
  - When the LSA describes a router, the link state ID is the OSPFv2 router ID of the router.
  - When an autonomous system external advertisement (Type 5) describes a default route, its link state ID is set to the default destination (0.0.0.0).
- ***ROUTER*** router or switch for which the command provides data. Options include:
  - <no parameter> all routers in the specified areas.
  - **adv-router** [*a.b.c.d*] an external router. Specifies local switch if an IP address is not included.
  - **self-originate** local switch. Equivalent to ***adv-router*** option without an IP address.

Examples

- This command displays the router link states contained in the area 2 LSDB.

```
switch>show ip ospf 1 2 database router

            OSPF Router with ID(192.168.103.1) (Process ID 1)

                 Router Link States (Area 0.0.0.2)

  LS age: 00:02:16
  Options: (E DC)
  LS Type: Router Links
  Link State ID: 192.168.103.1
  Advertising Router: 192.168.103.1
  LS Seq Number: 80000032
  Checksum: 0x1B60
  Length: 36
  Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 192.168.2.1
     (Link Data) Router Interface address: 192.168.2.1
      Number of TOS metrics: 0
       TOS 0 Metrics: 10


  LS age: 00:02:12
  Options: (E DC)
  LS Type: Router Links
  Link State ID: 192.168.104.2
  Advertising Router: 192.168.104.2
  LS Seq Number: 80000067
  Checksum: 0xA29C
  Length: 36
  Number of Links: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 192.168.2.1
     (Link Data) Router Interface address: 192.168.2.2
      Number of TOS metrics: 0
       TOS 0 Metrics: 10
switch>
```

- This command displays link state database (LSDB) contents for area 2.

```
switch>show ip ospf 1 2 database

            OSPF Router with ID(192.168.103.1) (Process ID 1)

                 Router Link States (Area 0.0.0.2)

Link ID         ADV Router      Age         Seq#       Checksum Link count
192.168.103.1   192.168.103.1   00:29:08    0x80000031 0x001D5F 1
192.168.104.2   192.168.104.2   00:29:09    0x80000066 0x00A49B 1

                 Net Link States (Area 0.0.0.2)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.2.1     192.168.103.1   00:29:08    0x80000001 0x00B89D

                 Summary Net Link States (Area 0.0.0.2)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.0.0     192.168.103.1   00:13:20    0x80000028 0x0008C8
192.168.0.0     192.168.104.2   00:09:16    0x80000054 0x00A2FF
192.168.3.0     192.168.104.2   00:24:16    0x80000004 0x00865F
192.168.3.0     192.168.103.1   00:24:20    0x80000004 0x002FC2
192.168.103.0   192.168.103.1   00:14:20    0x80000028 0x0096D2
192.168.103.0   192.168.104.2   00:13:16    0x80000004 0x00364B
192.168.104.0   192.168.104.2   00:08:16    0x80000055 0x002415
192.168.104.0   192.168.103.1   00:13:20    0x80000028 0x00EF6E
switch>
```

show ip ospf interface

The **show ip ospf interface** command displays interface information that is related to OSPFv2.

Command Mode

EXEC

Command Syntax

show ip ospf [*process_id*] **interface** [***INTERFACE_NAME***]

Parameters

- *process_id* process ID. Values range from 1 to 65535.
- ***INTERFACE_NAME*** Interface type and number. Values include
  - <no parameter> all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **port-channel** *p_num* Port channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Related Command

show ip ospf interface brief

Example

- This command displays complete OSPFv2 information for VLAN 1.

```
switch>show ip ospf interface vlan 1
Vlan1 is up, line protocol is up (connected)
  Internet Address 192.168.0.1/24, Area 0.0.0.0
  Process ID 1, Router ID 192.168.103.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router is 192.168.104.2
  Backup Designated router is 192.168.103.1
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 1
  MTU is 1500
switch>
```

In addition to displaying the IP address, area, and interval configuration, the display indicates that the switch is an ABR by displaying a neighbor count, the designated router, and backup designated router.

show ip ospf interface brief

The **show ip ospf interface brief** command displays a summary of OSPFv2 interfaces, states, addresses and masks, and areas on the router..

Command Mode

EXEC

Command Syntax

show ip ospf [*process_id*] **interface brief**

Parameters

- *process_id* OSPFv2 process ID. Values include:
  - <no parameter> Command returns data for all OSPFv2 instances.
  - 1 to 65535 Command returns data for specified OSPFv2 instance.

Related Commands

show ip ospf interface

Example

- This command displays a summary of interface information for the switch.

```
switch>show ip ospf interface brief
Interface    PID   Area            IP Address         Cost  State    Nbrs
Loopback0    1     0.0.0.0         192.168.103.1/24   10    DR       0
Vlan1        1     0.0.0.0         192.168.0.1/24     10    BDR      1
Vlan2        1     0.0.0.2         192.168.2.1/24     10    BDR      1
Vlan3        1     0.0.0.3         192.168.3.1/24     10    DR       0
switch>
```

Configuration information includes the process ID (PID), area, IP address, and cost. OSPFv2 operational information includes the designated router status and number of neighbors.

show ip ospf neighbor

The **show ip ospf neighbor** command displays OSPFv2 neighbor information for specified interfaces.

Command Mode

EXEC

Command Syntax

show ip ospf neighbor [***INTERFACE_NAME***] [***NEIGHBOR***] [***DATA_OPTION***]

Parameters

- ***INTERFACE_NAME*** Interface type and number. Values include:
  - <no parameter> all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* loopback interface specified by *l_num*.
  - **port-channel** *p_num* port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
- ***NEIGHBOR*** OSPFv2 neighbor. Options include:
  - <no parameter> all neighbors.
  - *ipv4_addr* Neighbor IPv4 address.
  - *host_name* Neighbor host name.
- ***DATA_OPTION*** Type of information the command displays. Values include:
  - <no parameter> Displays summary of all neighbors.
  - **adjacency-changes** Displays all adjacency changes.
  - **detail** Expands data to include DR and BDR addresses, adjacency establishment time, and other status.

Examples

- This command displays the switch's neighbors.

```
switch>show ip ospf neighbor
Neighbor ID     Pri   State        Dead Time   Address         Interface
192.168.104.2     1   FULL/DR      00:00:35    192.168.0.2     Vlan1
192.168.104.2     8   FULL/BDR     00:00:31    192.168.2.2     Vlan2
switch>
```

- This command displays details about the neighbors to VLAN 2.

```
switch>show ip ospf neighbor vlan 2 detail
 Neighbor 192.168.104.2, interface address 192.168.2.2
    In the area 0.0.0.2 via interface Vlan2
    Neighbor priority is 8, State is FULL, 13 state changes
    Adjacency was established 000:01:25:48 ago
    DR is 192.168.2.1 BDR is 192.168.2.2
    Options is E
    Dead timer due in 00:00:34
switch>
```

- This command displays the adjacency changes to VLAN 2.

```
switch>show ip ospf neighbor vlan 2 adjacency-changes
[08-04 08:55:32] 192.168.104.2, interface Vlan2 adjacency established
[08-04 09:58:51] 192.168.104.2, interface Vlan2 adjacency dropped: interface went
down
[08-04 09:58:58] 192.168.104.2, interface Vlan2 adjacency established
[08-04 09:59:34] 192.168.104.2, interface Vlan2 adjacency dropped: interface went
down
[08-04 09:59:42] 192.168.104.2, interface Vlan2 adjacency established
[08-04 10:01:40] 192.168.104.2, interface Vlan2 adjacency dropped: nbr did not
list our router ID
[08-04 10:01:46] 192.168.104.2, interface Vlan2 adjacency established
switch>
```

show ip ospf request-list

The **show ip ospf request-list** command displays a list of all OSPFv2 link state advertisements (LSAs) requested by a router.

Command Mode

EXEC

Command Syntax

```
show ip ospf request-list
```

Example

- This command displays an LSA request list.

```
switch>show ip ospf request-list
Neighbor 192.168.104.2 interface: 192.168.0.2 address vlan1
Type LS ID ADV RTR Seq No Age Checksum
Neighbor 192.168.104.2 interface: 192.168.2.2 address vlan2
Type LS ID ADV RTR Seq No Age Checksum
switch>
```

show ip ospf retransmission-list

The **show ip ospf retransmission-list** command displays a list of all OSPFv2 link state advertisements (LSAs) waiting to be re-sent.

Command Mode

EXEC

Command Syntax

show ip ospf retransmission-list

Example

- This command displays an empty retransmission list.

```
switch>show ip ospf retransmission-list
Neighbor 192.168.104.2 interface vlan1 address 192.168.0.2
LSA retransmission not currently scheduled. Queue length is 0

Type         Link ID      ADV Router   Age       Seq# Checksum
Neighbor 192.168.104.2 interface vlan2 address 192.168.2.2
LSA retransmission not currently scheduled. Queue length is 0

Type         Link ID      ADV Router   Age       Seq# Checksum
switch>
```

shutdown (OSPFv2)

The **shutdown** command disables OSPFv2 on the switch. Neighbor routers are notified of the shutdown and all traffic that has another path through the network will be directed to an alternate path.

OSPFv2 is disabled on individual interfaces with the **shutdown (OSPFv2)** command.

The **no shutdown** and **default shutdown** commands enable the OSPFv2 instance by removing the **shutdown** statement from the OSPF block in ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Examples

- This command disables OSPFv2 activity on the switch.

```
switch(config)#router ospf 6
switch(config-router-ospf)#shutdown
switch(config-router-ospf)#
```

- This command resumes OSPFv2 activity on the switch.

```
switch(config-router-ospf)#no shutdown
switch(config-router-ospf)#
```

timers spf (OSPFv2)

The **timers spf** command configures the shortest path first (SPF) timer. The SPF timer defines the maximum interval between OSPFv2 path calculations. The default period is five seconds.

The **no timers spf** and **default timers spf** commands restore the default maximum OSPFv2 path calculation interval to five seconds by removing the **timers spf** command from ***running-config***.

Command Mode

Router-OSPF Configuration

Command Syntax

```
timers spf spf_time
no timers spf
default timers spf
```

Parameters

- *spf_time* OSPFv2 path calculation interval (seconds). Values range from 1 to 65535. Default is 5.

Example

- This command sets the SPF timer to ten seconds.

```
switch(config)#router ospf 6
switch(config-router-ospf)#timers spf 10
switch(config-router-ospf)#
```

Chapter 20

Open Shortest Path First – Version 3

Open Shortest Path First (OSPF) is a link-state routing protocol that operates within a single autonomous system. OSPF version 3 is defined by RFC 5340.

This chapter contains the following sections.

- Section 20.1: OSPFv3 Introduction
- Section 20.2: OSPFv3 Conceptual Overview
- Section 20.3: Configuring OSPFv3
- Section 20.4: OSPFv3 Examples
- Section 20.5: OSPFv3 Commands

20.1 OSPFv3 Introduction

OSPFv3 is based on OSPF version 2 and includes enhancements that utilize IPv6 features. However, OSPFv3 is configured and operates independently of any implementation of OSPFv2 on the switch. OSPFv2 features that OSPFv3 implements include:

- Packet types
- Neighbor discovery and adjacency formation mechanisms
- LSA aging and flooding
- SPF calculations
- DR election procedure
- Multiple area support
- Router-ID (32 bits)

The following list describes the OSPFv3 differences and enhancements from OSPFv2:

- IPv6 128-bit addresses
- Use of link-local addresses
- OSPFv3 runs over links instead of subnets

Arista switches support these OSPFv3 functions:

- A single OSPFv3 instance
- Intra- and inter-area routing
- Type 1 and 2 external routing
- Broadcast and P2P interfaces
- Stub areas
- Redistribution of static and connected routes into OSPFv3
- Largely industry standard compatible CLI

20.2 OSPFv3 Conceptual Overview

20.2.1 Storing Link States

OSPFv3 is a dynamic, link-state routing protocol, where links represent routable paths. Dynamic routing protocols calculate the most efficient path between locations based on bandwidth and device status.

A link state advertisement (LSA) is an OSPFv3 packet that communicates a router's topology to other routers. The link state database (LSDB) stores an area's topology database and is composed of LSAs received from other routers. Routers update the LSDB by storing LSAs from other routers.

20.2.2 Topology

An autonomous system (AS) is the IP domain where a dynamic protocol routes traffic. In OSPFv3, an AS is composed of areas, which define the LSDB computation boundaries. All routers in an area store identical LSDBs. Routers in different areas exchange updates without storing the entire database, reducing information maintenance on large, dynamic networks.

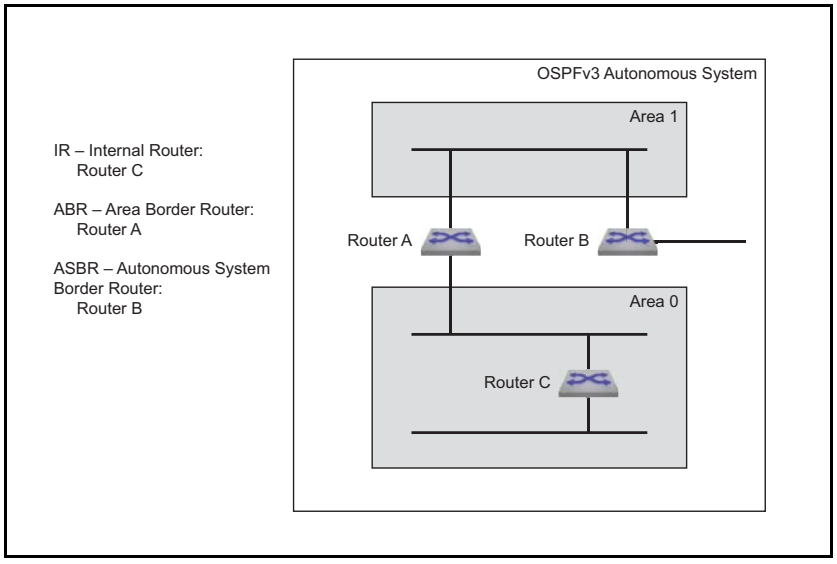
An AS shares internal routing information from its areas and external routing information from other processes to inform routers outside the AS about routes the network can access. Routers that advertise routes on other ASs commit to carry data to the IP space on the route.

OSPFv3 defines these routers:

- Internal router (IR) – a router whose interfaces are contained in a single area. All IRs in an area maintain identical LSDBs.
- Area border router (ABR) – a router that has interfaces in multiple areas. ABRs maintain one LSDB for each connected area.
- Autonomous system boundary router (ASBR) – a gateway router connecting the OSPFv3 domain to external routes, including static routes and routes from other autonomous systems.

Figure 20-1 displays the OSPFv3 router types.

Figure 20-1 OSPFv3 Router Types



OSPFv3 areas are assigned a number between 0 and 4,294,967,295. Area numbers are often expressed in dotted decimal notation, similar to IP addresses.

Each AS has a backbone area, designated as area 0, that connects to all other areas. The backbone receives routing information from all areas, then distributes it to the other areas as required.

OSPFv3 area types include:

- Normal area – accepts intra-area, inter-area, and external routes. The backbone is a normal area.
- Stub area – does not receive router advertisements external to the AS. Stub area routing is based on a default route.

20.2.3 Link Updates

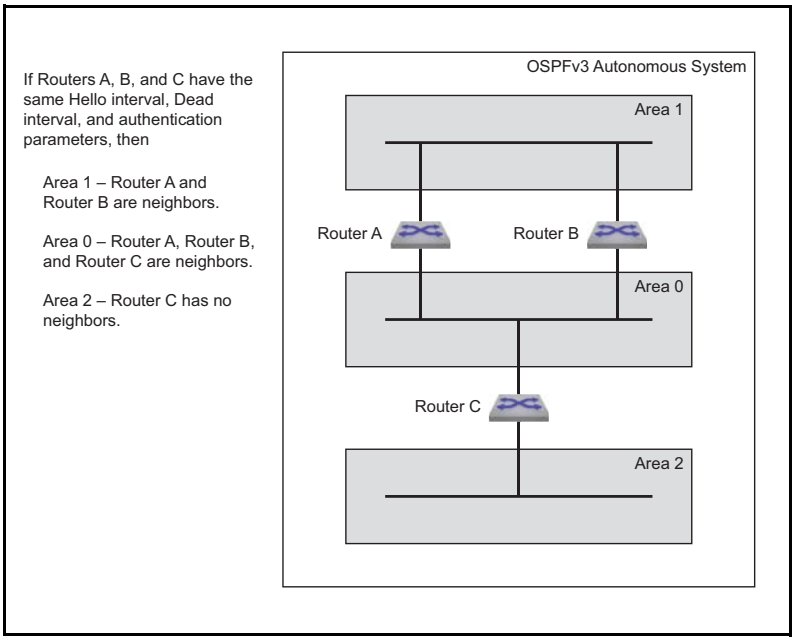
Routers periodically send hello packets to advertise status and establish neighbors. A router's hello packet includes IP addresses of other routers from which it received a hello packet within the time specified by the router dead interval. Routers become neighbors when they detect each other in their hello packets if they:

- share a common network segment.
- are in the same area.
- have the same hello interval, dead interval, and authentication parameters.

Neighbors form adjacencies to exchange LSDB information. A neighbor group uses hello packets to elect a Designated Router (DR) and Backup Designated Router (BDR). The DR and BDR become adjacent to all other neighbors, including each other. Only adjacent neighbors share database information.

Figure 20-2 illustrates OSPFv3 neighbors.

Figure 20-2 OSPFv3 Neighbors



The DR is the central contact for database exchanges. Switches send database information to their DR, which relays the information to the other neighbors. All routers in an area maintain identical LSDBs. Switches also send database information to their BDR, which stores this data without distributing it. If the DR fails, the BDR distributes LSDB information to its neighbors.

OSPFv3 routers distribute LSAs by sending them on all of their active interfaces. Passive interfaces send LSAs to active interfaces but do not receive LSAs, thus alerting OSPFv3 routers of devices that do not otherwise participate in OSPFv3. The router does not send or process OSPFv3 packets received on passive interfaces, including hello packets, which causes the interface to drop its adjacencies.

When a router's LSDB is changed by an LSA, it sends the changes to the DR and BDR for distribution to the other neighbors. Routing information is updated only when the topology changes.

Routing devices use Dijkstra's algorithm to calculate the shortest path to all known destinations, based on cumulative route cost. The cost of an interface indicates the transmission overhead and is usually inversely proportional to its bandwidth.

20.3 Configuring OSPFv3

These sections describe basic OSPFv3 configuration steps:

- Section 20.3.1: Configuring the OSPFv3 Instance
- Section 20.3.2: Configuring OSPFv3 Areas
- Section 20.3.3: Configuring Interfaces for OSPFv3
- Section 20.3.4: Enabling OSPFv3
- Section 20.3.5: Displaying OSPFv3 Status

20.3.1 Configuring the OSPFv3 Instance

20.3.1.1 Entering OSPFv3 Configuration Mode

OSPFv3 configuration commands apply to the OSPFv3 instance. To perform OSPFv3 configuration commands, the switch must be in router-ospf3 configuration mode. The **ipv6 router ospf** command places the switch in router-ospf3 configuration mode, creating an OSPFv3 instance if OSPF3 was not previously instantiated on the switch.

The process ID identifies the OSPFv3 instance and is local to the router. Neighbor OSPFv3 routers can have different process IDs.

The switch supports one OSPFv3 instance. When an OSPFv3 instance exists, the **ipv6 router ospf** command must specify its process ID. Attempts to define additional instances will generate errors. The **show ipv6 ospf** command displays information the OSPF instance, including its process ID.

Example

- This command places the switch in router-ospf3 configuration mode. If OSPF was not previously instantiated, the command creates an OSPFv3 instance with a process ID of 9.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

20.3.1.2 Defining the Router ID

The router ID is a 32-bit number assigned to a router running OSPFv3. This number uniquely labels the router within an Autonomous System. Status commands identify the switch through the router ID.

The switch sets the router ID to the first available alternative in the following list:

1. The **router-id** command.
2. The loopback IPv6 address, if a loopback interface is active on the switch.
3. The highest IPv6 address on the router.

The **router-id (OSPFv3)** command configures the router ID for an OSPFv3 instance.

Example

- This command assigns 15.1.1.1 as the OSPFv3 router ID.

```
switch(config-router-ospf3)#router-id 15.21.4.9
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   router-id 15.21.4.9
switch(config-router-ospf3)#
```

20.3.1.3 Global OSPFv3 Parameters

These router-ospf3 configuration mode commands define OSPFv3 behavior.

Logging Adjacency Changes

The **log-adjacency-changes (OSPFv3)** command configures the switch to send a syslog message when a neighbor goes up or down.

Examples

- This command configures the switch to send a syslog message when an OSPFv3 neighbor goes up or down.

```
switch(config-router-ospf3)#log-adjacency-changes
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   router-id 15.21.4.9
   log-adjacency-changes
switch(config-router-ospf3)#
```

Intra-Area Distance

The **distance ospf intra-area (OSPFv3)** command configures the administrative distance for routes contained in a single OSPFv3 area. Administrative distances compare dynamic routes configured by different protocols. The default administrative distance for intra-area routes is 10.

Example

- This command configures an administrative distance of 90 for OSPFv3 intra-area routes.

```
switch(config-router-ospf3)#distance ospf intra-area 90
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   distance ospf intra-area 90
switch(config-router-ospf3)#
```

Passive Interfaces

The **passive-interface (OSPFv3)** command prevents the transmission of hello packets on the specified interface. Passive interfaces drop all adjacencies and do not form new adjacencies. Although passive interfaces do not send or receive LSAs, other interfaces may generate LSAs for the network segment. The router does not send or process OSPFv3 packets received on passive interfaces. The router advertises the passive interface in the router LSA.

The **no passive-interface** command re-enables OSPFv3 processing on the specified interface.

Examples

- This command configures VLAN 200 as a passive interface.

```
switch(config-router-ospf3)#passive-interface vlan 200
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   passive-interface Vlan200
switch(config-router-ospf3)#
```

- This command configures VLAN 200 as an active interface.

```
switch(config-router-ospf3)#no passive-interface vlan 200
switch(config-router-ospf3)#show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

Redistributing Connected Routes

Redistributing connected routes causes the OSPFv3 instance to advertise all connected routes on the switch as external OSPFv3 routes. Connected routes are routes that are established when IPv6 is enabled on an interface.

Example

- The **redistribute (OSPFv3) connected** command converts connected routes to OSPFv3 external routes.

```
switch(config-router-ospf3)#redistribute connected
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   redistribute connected
switch(config-router-ospf3)#
```

Redistributing Static Routes

Redistributing static routes causes the OSPFv3 instance to advertise all static routes on the switch as external OSPFv3 routes. The switch does not support redistributing individual static routes.

Example

- The **redistribute (OSPFv3) static** command converts the static routes to OSPFv3 external routes.

```
switch(config-router-ospf3)#redistribute static
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   redistribute static
switch(config-router-ospf3)#
```

20.3.2 Configuring OSPFv3 Areas

OSPFv3 areas are configured through area commands. The switch must be in router-ospf3 configuration mode, as described in Section 20.3.1.1: Entering OSPFv3 Configuration Mode, to run area commands.

Areas are assigned a 32-bit number that is expressed in decimal or dotted-decimal notation. When an OSPFv3 instance spans multiple routers, the switch only configures areas that connect to its interfaces.

20.3.2.1 Configuring the Area Type

The **area (OSPFv3)** command specifies the area type. The switch supports three area types:

- Normal area: Area that accepts intra-area, inter-area, and external routes. The backbone area (area 0) is a normal area.
- Stub area: Area where external routes are not advertised. External routes are reached through a default summary route (0.0.0.0) inserted into stub areas. Networks with no external routes do not require stub areas.

The default area type is normal.

Example

- These commands configures area 200 as a normal area and 300 as a stub area.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#area 200
switch(config-router-ospf3)#area 300 stub
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   area 0.0.0.200
   area 0.0.1.44 stub
switch(config-router-ospf3)#
```

20.3.2.2 Configuring Area Parameters

These router-ospf3 configuration mode commands define OSPFv3 behavior in a specified area.

Default Summary Route Cost

The **area default-cost (OSPFv3)** command specifies the cost of the default summary route that ABRs send into a stub area or NSSA. Summary routes, also called inter-area routes, originate in areas different than their destination. When the **area default-cost** command is not configured for an area, the default-cost of that area is set to 10.

Example

- This command configures a cost of 25 for the default summary route in area 0.0.1.194 (450).

```
switch(config-router-ospf3)#area 450 default-cost 25
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   area 0.0.1.194 default-cost 25
```

20.3.3 Configuring Interfaces for OSPFv3

OSPFv3 interface configuration commands enable OSPFv3 on an interface, assign the interface to an area, and specify transmission parameters for routed ports and SVIs that handle OSPFv3 packets.

20.3.3.1 Assigning an Interface to an Area

The **ipv6 ospf area** command enables OSPFv3 on the configuration mode interface and associates the specified area to the interface. Each routed interface can be associated with one OSPFv3 area; subsequent **ipv6 ospf area** commands that designate a different area on an interface replace any existing command for the interface.

Example

- These commands enable OSPFv3 on VLAN interface 200 and associates area 0 to the interface.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf 9 area 0
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf 9 area 0.0.0.0
switch(config-if-Vl200)#
```

20.3.3.2 Configuring Intervals

Interval configuration commands determine OSPFv3 packet transmission characteristics for a specfied VLAN interface. Interval configuration commands are entered in vlan-interface configuration mode.

Hello Interval

The hello interval specifies the period between consecutive hello packet transmissions from an interface. Each OSPFv3 neighbor should specify the same hello interval, which should not be longer than any neighbor's dead interval.

The **ipv6 ospf hello-interval** command configures the hello interval for the configuration mode interface. The default is 10 seconds.

Example

- These commands configure a hello interval of 45 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf hello-interval 45
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf hello-interval 45
switch(config-if-Vl200)#
```

Dead Interval

The dead interval specifies the period that an interface waits for an OSPFv3 packet from a neighbor before it disables the adjacency under the assumption that the neighbor is down. The dead interval should be configured identically on all OSPFv3 neighbors and be longer than the hello interval of any neighbor.

The **ipv6 ospf dead-interval** command configures the dead interval for the configuration mode interface. The default is 40 seconds.

Example

- This command configures a dead interval of 75 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf dead-interval 75
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf dead-interval 75
switch(config-if-Vl200)#
```

Retransmission Interval

Routers that send OSPFv3 advertisements to an adjacent router expect to receive an acknowledgment from that neighbor. Routers that do not receive an acknowledgment will retransmit the advertisement. The retransmission interval specifies the period between retransmissions.

The **ipv6 ospf retransmit-interval** command configures the LSA retransmission interval for the configuration mode interface. The default retransmission interval is 5 seconds.

Example

- This command configures a retransmission interval of 25 seconds for VLAN 200..

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf retransmit-interval 25
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf retransmit-interval 25
switch(config-if-Vl200)#
```

Transmission Delay

The transmission delay is an estimate of the time that an interface requires to transmit a link-state update packet. OSPFv3 adds this delay to the age of outbound packets to more accurately reflect the age of the LSA when received by a neighbor.

The **ipv6 ospf transmit-delay** command configures the transmission delay for the configuration mode interface. The default transmission delay is one second.

Example

- This command configures a transmission delay of 10 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf transmit-delay 10
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf transmit-delay 10
switch(config-if-Vl200)#
```

20.3.3.3 Configuring Interface Parameters

Interface Cost

The OSPFv3 interface cost reflects the overhead of sending packets across the interface. The cost is typically assigned to be inversely proportional to the bandwidth of the interface. The **ipv6 ospf cost** command configures the OSPFv3 cost for the configuration mode interface. The default cost is 10.

Example

- This command configures a cost of 50 for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf cost 50
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf cost 50
switch(config-if-Vl200)#
```

Router Priority

Router priority determines preference during designated router (DR) and backup designated router (BDR) elections. Routers with higher priority numbers have preference over other routers. Routers with a priority of zero cannot be elected as a DR or BDR.

The **ipv6 ospf priority** command configures router priority for the configuration mode interface. The default priority is 1.

Example

- This command configures a router priority of 128 for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf priority 128
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf priority 128
switch(config-if-Vl200)#
```

20.3.4 Enabling OSPFv3

20.3.4.1 IP Routing

OSPFv3 requires that IPv6 unicast routing is enabled on the switch. When IP routing is not enabled, entering OSPFv3 configuration mode generates a message.

Examples

- This message is displayed if, when entering router-ospf3 configuration mode, IPv6 unicast routing is not enabled.

```
switch(config)#ipv6 router ospf 9
! IPv6 routing not enabled
switch(config-router-ospf3)#
```

- This command enables IP routing on the switch.

```
switch(config)#ipv6 unicast-routing
```

20.3.4.2 Disabling OSPFv3

The **shutdown (OSPFv3)** disables OSPFv3 operations on the without disrupting the OSPFv3 configuration. To disable OSPFv3 on an interface, remove the **ipv6 ospf area** statement for the corresponding interface.

The **no shutdown** command resumes OSPFv3 activity.

Examples

- This command disables OSPFv3 activity on the switch.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#shutdown
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   shutdown
switch(config-router-ospf3)#
```

- This command resumes OSPFv3 activity on the switch.

```
switch(config-router-ospf3)#no shutdown
switch(config-router-ospf3)#show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

20.3.5 Displaying OSPFv3 Status

This section describes OSPFv3 **show** commands that display OSPFv3 status. General switch methods that provide OSPFv3 information include pinging routes, viewing route status (**show ip route** command), and viewing the configuration (**show running-config** command).

20.3.5.1 OSPFv3 Summary

The **show ipv6 ospf** command displays general OSPFv3 configuration information and operational statistics. The output lists configuration parameters and operational statistics and status for the OSPFv3 instance, followed by a brief description of the areas located on the switch.

Example

- This command displays OSPFv3 routing process information..

```
switch>show ipv6 ospf
Routing Process "ospfv3 1" with ID 171.37.0.23 and Instance 0
  It is an autonomous system boundary router and is an area border router
  Hold time between two SPFs is 5
  Minimum LSA interval 5. Minimum LSA arrival 1
  It has 13 fully adjacent neighbors
  Number of areas in this router is 2. 2 normal, 0 stub, 0 nssa
  Graceful restart is enabled
    Grace period is 40
    Strict helper is enabled
  SPF algorithm last executed 00:02:59 ago
  Area 0.0.0.0
    Number of interface in this area is 8
    It is a normal area
  Area 0.0.0.2
    Number of interface in this area is 1
    It is a normal area
```

20.3.5.2 Viewing OSPFv3 on the Interfaces

The **show ipv6 ospf interface** command displays OSPFv3 information for switch interfaces configured for OSPFv3. Different command options allow the display of either all interfaces or a specified interface. The command can also be configured to display complete information or a brief summary.

Example

- This command displays OSPFv3 information for interfaces where OSPFv3 is enabled.

```
switch#show ipv6 ospf interface
Ethernet17 is up
  Interface Address fe80::48c:73ff:fe00:1319%Ethernet12, Area 0.0.0.0
  Network Type Broadcast, Cost 10
  Transmit Delay is 1 sec, State Backup DR, Priority 1
  Designated Router is 171.37.0.37
  Backup Designated Router is 171.37.0.23
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 1
Vlan31 is up
  Interface Address fe80::48c:73ff:fe00:1319%Vlan31, Area 0.0.0.0
  Network Type Broadcast, Cost 10
  Transmit Delay is 1 sec, State Backup DR, Priority 1
  Designated Router is 171.37.0.22
  Backup Designated Router is 171.37.0.23
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 1
Vlan32 is up
  Interface Address fe80::48c:73ff:fe00:1319%Vlan32, Area 0.0.0.0
  Network Type Broadcast, Cost 10
  Transmit Delay is 1 sec, State DR Other, Priority 1
  Designated Router is 171.37.0.11
  Backup Designated Router is 171.37.0.22
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 2
switch#
```

20.3.5.3 Viewing the OSPFv3 Database

The **show ipv6 ospf database** command displays the LSAs in the LSDB for the specified area. If no area is listed, the command displays the contents of the database for each area on the switch. The database command provides options to display subsets of the LSDB database, a summary of database contents, and the link states that comprise the database.

Example

- This command displays the OSPFv3 database of link state advertisements (LSAs).

```
switch#show ipv6 ospf database
Routing Process "ospf 9":

  AS Scope LSDB

Type         Link ID     ADV Router  Age        Seq#   Checksum
 AEX         0.0.0.5    171.37.0.37   15 0x80000005   0x00be82
 AEX         0.0.0.9    171.37.0.22 1747 0x8000002b   0x00df56
 AEX         0.0.0.3    171.37.0.46  599 0x8000002d   0x00651d
<-------OUTPUT OMITTED FROM EXAMPLE-------->


  Area 0.0.0.0 LSDB

Type         Link ID     ADV Router  Age        Seq#   Checksum
 RTR         0.0.0.0    171.37.0.32  234 0x80000031   0x00585a
 NTW        0.0.0.26    171.37.0.32  271 0x80000005   0x005609
 NAP        0.0.0.26    171.37.0.32  274 0x80000005   0x00964c
<-------OUTPUT OMITTED FROM EXAMPLE-------->


  Interface vlan3911 LSDB

Type         Link ID     ADV Router  Age        Seq#   Checksum
 LNK        0.0.0.38    171.37.0.22  267 0x80000005   0x00a45a
 LNK        0.0.0.23    171.37.0.23  270 0x8000002c   0x005b7e

  Interface vlan3902 LSDB

Type         Link ID     ADV Router  Age        Seq#   Checksum
 LNK        0.0.0.17    171.37.0.11 1535 0x8000002b   0x007120
 LNK        0.0.0.37    171.37.0.22    7 0x8000002b   0x00ce23
 LNK        0.0.0.22    171.37.0.23  250 0x8000002d   0x00c350
<-------OUTPUT OMITTED FROM EXAMPLE-------->


switch#
```

20.3.5.4 Viewing OSPFv3 Neighbors

The **show ipv6 ospf neighbor** command displays information about the routers that are neighbors to the switch. Command options allow the display of summary or detailed information about the neighbors to all areas and interfaces on the switch. The command also allows for the display of neighbors to individual interfaces or areas. The ***adjacency-changes*** option displays the interface's adjacency changes.

Example

- This command displays the switch's neighbors.

```
switch#show ipv6 ospf neighbor
Routing Process "ospf 9":
Neighbor 171.37.0.37 priority is 1, state is Full
  In area 0.0.0.0 interface et12
  DR is 171.37.0.37 BDR is 171.37.0.23
  Options is 0
  Dead timer is due in 37 seconds
Neighbor 171.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3911
  DR is 171.37.0.22 BDR is 171.37.0.23
  Options is 0
  Dead timer is due in 31 seconds
Neighbor 171.37.0.11 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3902
  DR is 171.37.0.11 BDR is 171.37.0.22
  Options is 0
  Dead timer is due in 33 seconds
Neighbor 171.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3902
  DR is 171.37.0.11 BDR is 171.37.0.22
  Options is 0
  Dead timer is due in 31 seconds
Neighbor 171.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3908
  DR is 171.37.0.22 BDR is 171.37.0.21
  Options is 0
  Dead timer is due in 39 seconds

switch#
```

20.3.5.5 Viewing OSPFv3 Routes

The **show ipv6 routes** command provides an OSPFv3 option.

Example

- This command displays the switch's OSPFv3 routes.

```
switch#show ipv6 route ospf
IPv6 Routing Table - 43 entries
Codes: C - connected, S - static, K - kernel, O - OSPF, B - BGP, R - RIP,
A - Aggregate

O   fd7a:3279:81a4:1112::/64 [150/11]
     via fe80::21c:41ff:fe00:d120, Ethernet12
O   fd7a:3279:81a4:1114::/64 [150/11]
     via fe80::21c:41ff:fe00:d120, Ethernet12
O   fd7a:3279:81a4:1124::/64 [10/20]
     via fe80::21c:41ff:fe01:5fe1, Vlan3901
     via fe80::21c:41ff:fe01:5fe1, Vlan3902
     via fe80::21c:41ff:fe01:5fe1, Vlan3908
O   fd7a:3279:81a4:1a00::25/128 [150/11]
     via fe80::21c:41ff:fe00:d120, Ethernet12
O   fd7a:3279:81a4:1a00::28/128 [150/11]
     via fd7a:3279:81a4:fe40::5, Vlan3908

switch#
```

20.4 OSPFv3 Examples

This section describes the commands required to configure three OSPFv3 topologies.

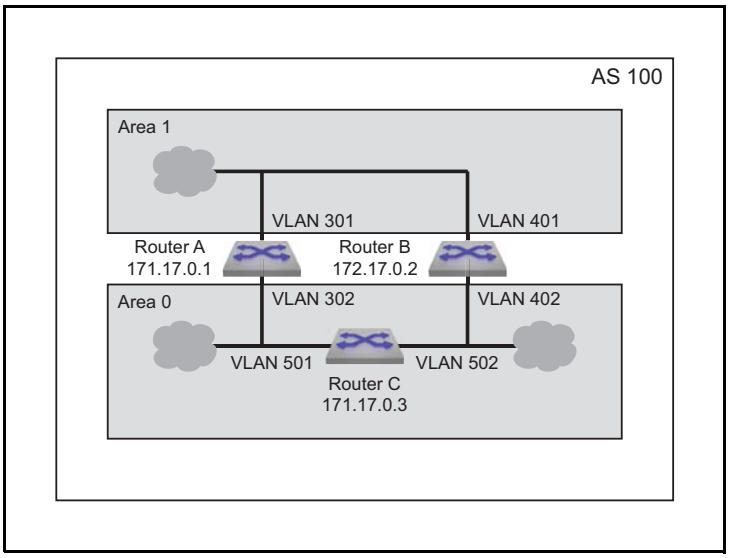
20.4.1 OSPFv3 Example 1

The AS in Example 1 contains two areas that are connected through two routers. The backbone area also contains an internal router that connects two links.

20.4.1.1 Example 1 Diagram

Figure 20-3 displays the Example 1 topology. Two ABRs connect area 0 and area 1 – Router A and Router B. Router C is an internal router that connects two links in area 0. Area 0 is normal; area is stub.

Figure 20-3 OSPFv3 Example 1



Area 1 Configuration

Area 1 contains links to ABRs Router A and Router B.

- Router A is accessed through VLAN 301.
- Router B is accessed through VLAN 401.
- Designated Router (DR): Router A.
- Backup Designated Router (BDR): Router B.
- Each router defines an interface cost of 10.
- Router priority is not specified for either router on area 1.

Area 0 ABR Configuration

Area 0 contains links to ABRs Router A and Router B.

- Router A is accessed through VLAN 302.
- Router B is accessed through VLAN 402.
- Designated Router (DR): Router B.
- Backup Designated Router (BDR): Router A.
- Each router defines an interface cost of 20.
- Each router defines a retransmit-interval of 10.

- Each router defines a transmit-delay of 2.
- Router priority is specified such that Router B will be elected as the Designated Router.

Area 0 IR Configuration

Area 0 contains two links to an internal router.

- Router C is accessed through VLAN 501 and VLAN 502.
- VLAN 501 is configured as follows:
  - Interface cost of 20.
  - Retransmit-interval of 10.
  - Transmit-delay of 2.
- VLAN 502 is configured as follows:
  - Interface cost of 20.
  - Dead interval of 80 seconds.

20.4.1.2 Example 1 Code

This code configures the OSPFv3 instances on the three switches.

Step 1 Configure the areas and router IDs.

Step a Router A OSPFv3 instance configuration:

```
switch-A(config)#ipv6 router ospf 100
switch-A(config-router-ospfv3)#area 1 stub
switch-A(config-router-ospfv3)#router-id 171.17.0.1
```

Step b Router B OSPFv3 instance configuration::

```
switch-B(config)#ipv6 router ospf 100
switch-B(config-router-ospfv3)#area 1 stub
switch-B(config-router-ospfv3)#router-id 171.17.0.2
```

Step c Router C OSPFv3 instance configuration: interfaces:

```
switch-C(config)#ipv6 router ospf 100
switch-C(config-router-ospfv3)#router-id 171.17.0.3
```

Step 2 Configure the interface OSPFv3 area and transmission parameters.

Step a Router A interfaces:

```
switch-A(config)#interface vlan 301
switch-A(config-if-Vl301)#ipv6 ospf 100 area 1
switch-A(config-if-Vl301)#ip ospf cost 10
switch-A(config-if-Vl301)#ip ospf priority 6
switch-A(config-if-Vl301)#exit
switch-A(config)#interface vlan 302
switch-A(config-if-Vl302)#ipv6 ospf 100 area 0
switch-A(config-if-Vl302)#ip ospf cost 20
switch-A(config-if-Vl302)#ip ospf retransmit-interval 10
switch-A(config-if-Vl302)#ip ospf transmit-delay 2
switch-A(config-if-Vl302)#ip ospf priority 4
```

Step b Router B interfaces:

```
switch-B(config)#interface vlan 401
switch-B(config-if-Vl401)#ipv6 ospf 100 area 1
switch-B(config-if-Vl401)#ip ospf cost 10
switch-B(config-if-Vl401)#ip ospf priority 4
switch-B(config-if-Vl401)#exit
switch-B(config)#interface vlan 402
switch-B(config-if-Vl402)#ipv6 ospf 100 area 0
switch-B(config-if-Vl402)#ip ospf cost 20
switch-B(config-if-Vl402)#ip ospf retransmit-interval 10
switch-B(config-if-Vl402)#ip ospf transmit-delay 2
switch-B(config-if-Vl402)#ip ospf priority 6
```

Step c Router C interfaces:

```
switch-C(config)#interface vlan 501
switch-C(config-if-Vl501)#ipv6 ospf 100 area 0
switch-C(config-if-Vl501)#ip ospf cost 20
switch-C(config-if-Vl501)#ip ospf retransmit-interval 10
switch-C(config-if-Vl501)#ip ospf transmit-delay 2
switch-C(config-if-Vl501)#exit
switch-C(config)#interface vlan 502
switch-C(config-if-Vl502)#ipv6 ospf 100 area 0
switch-C(config-if-Vl502)#ip ospf cost 20
switch-C(config-if-Vl502)#ip ospf dead-interval 80
```

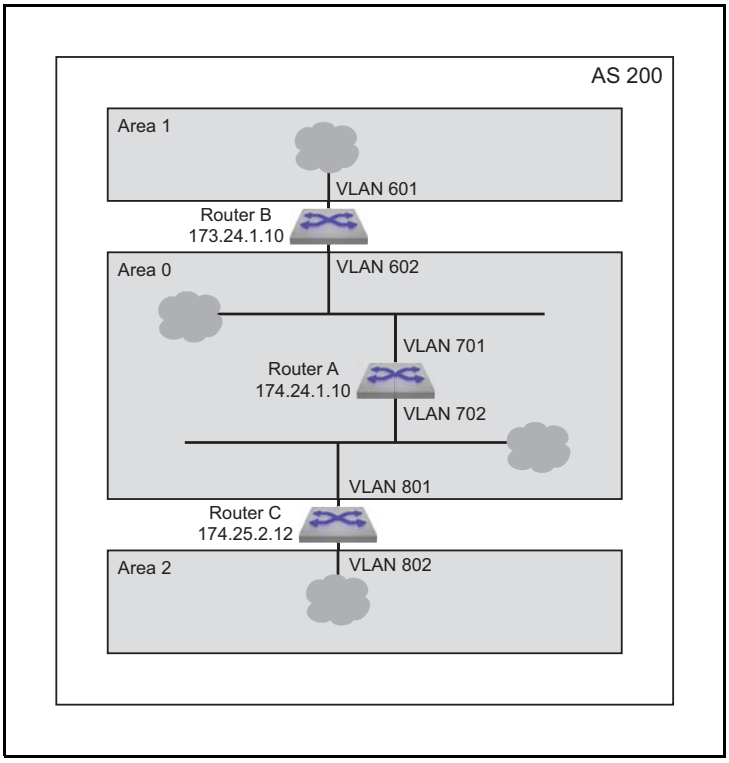
20.4.2 OSPFv3 Example 2

The AS in Example 2 contains three areas. Area 0 connects to the other areas through different routers and contains an internal router connecting two links. Area 0 is normal; the other areas are stub areas.

20.4.2.1 Example 2 Diagram

Figure 20-4 displays the Example 2 topology. One ABR (Router B) connects area 0 and area 1; another ABR (router C) connects area 0 and area 2. Router A is an internal router that connects two links in area 0.

Figure 20-4 OSPFv3 Example 2



Area 1 Configuration

Area 1 contains one link that is accessed by Router B.

- Router B is accessed through VLAN 601.
- The router defines a interface cost of 10.

Area 2 Configuration

Area 2 contains one link that is accessed by Router C.

- Router C is accessed through VLAN 802.
- The router defines a interface cost of 20.

Area 0 ABR Configuration

One ABR Router B link connects area 1 to area 0. One ABR Router C link connects area 0 to area 2.

- Router B is accessed through VLAN 602.
- Router C is accessed through VLAN 801.
- Designated Router (DR): Router B.

- Backup Designated Router (BDR): Router C.
- Each router defines an interface cost of 20.
- Each router defines a retransmit-interval of 10.
- Each router defines a transmit-delay of 2.

Area 0 IR Configuration

Area 0 contains links connected by an internal router.

- Router A is accessed through VLAN 701 and 702.
- The VLAN 701 link is configured as follows:
  — Interface cost of 10.
- The VLAN 702 link is configured as follows:
  — Interface cost of 20.
  — Retransmit-interval of 10.
  — Transmit-delay of 2.

20.4.2.2 Example 2 Code

Step 1 Configure the areas and router IDs.

Step a Router A OSPFv3 instance configuration:

```
switch-A(config)#ipv6 router ospf 200
switch-A(config-router-ospfv3)#router-id 174.24.1.10
```

Step b Router B OSPFv3 instance configuration:

```
switch-B(config)#ipv6 router ospf 200
switch-B(config-router-ospfv3)#area 1 stub
switch-B(config-router-ospfv3)#router-id 173.24.1.10
```

Step c Router C OSPFv3 instance configuration:

```
switch-C(config)#ipv6 router ospf 200
switch-C(config-router-ospfv3)#area 1 stub
switch-C(config-router-ospfv3)#router-id 174.25.2.12
```

Step 2 Configure the interface OSPFv3 area and transmission parameters.

Step a Router A interfaces:

```
switch-A(config)#interface vlan 701
switch-A(config-if-Vl701)#ipv6 ospf 200 area 0
switch-A(config-if-Vl701)#ip ospf cost 10
switch-A(config-if-Vl701)#exit
switch-A(config)#interface vlan 702
switch-A(config-if-Vl702)#ipv6 ospf 200 area 0
switch-A(config-if-Vl702)#ip ospf cost 20
switch-A(config-if-Vl702)#ip ospf retransmit-interval 10
switch-A(config-if-Vl702)#ip ospf transmit-delay 2
```

Step b Router B interfaces:

```
switch-B(config)#interface vlan 601
switch-B(config-if-Vl601)#ipv6 ospf 200 area 1
switch-B(config-if-Vl601)#ip ospf cost 10
switch-B(config-if-Vl601)#exit
switch-B(config)#interface vlan 602
switch-B(config-if-Vl602)#ipv6 ospf 200 area 0
switch-B(config-if-Vl602)#ip ospf cost 20
switch-B(config-if-Vl602)#ip ospf retransmit-interval 10
switch-B(config-if-Vl602)#ip ospf transmit-delay 2
switch-B(config-if-Vl602)#ip ospf priority 6
```

Step c Router C interfaces:

```
switch-C(config)#interface vlan 801
switch-C(config-if-Vl801)#ipv6 ospf 200 area 0
switch-C(config-if-Vl801)#ip ospf cost 20
switch-C(config-if-Vl801)#ip ospf retransmit-interval 10
switch-C(config-if-Vl801)#ip ospf transmit-delay 2
switch-C(config-if-Vl801)#exit
switch-C(config)#interface vlan 802
switch-C(config-if-Vl802)#ipv6 ospf 200 area 2
switch-C(config-if-Vl802)#ip ospf cost 20
switch-C(config-if-Vl802)#ip ospf dead-interval 80
```

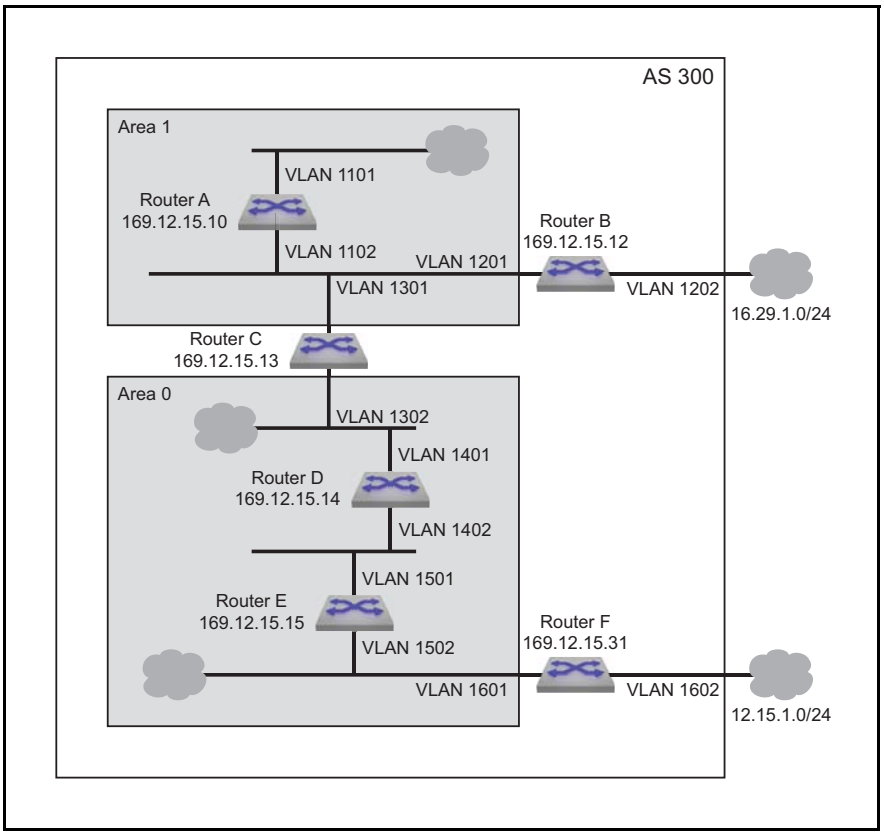
20.4.3 OSPFv3 Example 3

The AS in Example 3 contains two areas that connect through one ABR. Each area also contains an ASBR that connects static routes to the AS.

20.4.3.1 Example 3 Diagram

Figure 20-5 displays the Example 3 topology. One ABR connects area 0 and area 1. Router C is an ABR that connects the areas. Router A is an internal router that connects two links in area 1. Router D and Router E are internal routers that connect links in area 0. Router B and Router F are ASBRs that connect static routes outside the AS to area 1 and area 0, respectively.

Figure 20-5 OSPFv3 Example 3



Area 0 ABR Configuration

ABR Router C connects one area 0 link to an area 1 link.

- Router C is accessed through VLAN 1302.
- All interface OSPFv3 parameters are set to their default values.

Area 0 IR Configuration

Area 0 contains two internal routers, each of which connects two of the three links in the area.

- Router D is accessed through VLAN 1401 and VLAN 1402.
- Router E is accessed through VLAN 1501 and VLAN 1502.
- All interface OSPFv3 parameters are set to their default values.

Area 0 ASBR Configuration

ASBR Router F connects one area 0 link to an external link.

- Router F is accessed through VLAN 1601.
- Router F connects to the external AS through VLAN 1602.
- All interface OSPFv3 parameters are set to their default values.

Area 1 ABR Configuration

ABR Router C connects one area 0 link to an area 1 link.

- Router C is accessed by area 1 through VLAN 1301.
- Router C is accessed by area 0 through VLAN 1302.
- All interface OSPFv3 parameters are set to their default values.

Area 1 IR Configuration

Area 1 contains one internal router that connects two links in the area.

- Router A is accessed through VLAN 1101 and VLAN 1102.
- All interface OSPFv3 parameters are set to their default values.

Area 1 ASBR Configuration

ASBR Router B connects one area 1 link to an external link.

- Router B is access through VLAN 1201.
- Router B connects to the external AS through VLAN 1202.
- All interface OSPFv3 parameters are set to their default values.

20.4.3.2 Example 3 Code

Step 1 Configure the areas and router IDs.

Step a Router A OSPFv3 instance configuration:

```
switch-A(config)#ipv6 router ospf 300
switch-A(config-router-ospfv3)#router-id 169.12.15.10
switch-A(config-router-ospfv3)#area 1 stub
```

Step b Router B OSPFv3 instance configuration:

```
switch-B(config)#ipv6 router ospf 300
switch-B(config-router-ospfv3)#router-id 169.12.15.12
switch-B(config-router-ospfv3)#area 1 stub
```

Step c Router OSPFv3 instance configuration:

```
switch-C(config)#ipv6 router ospf 300
switch-C(config-router-ospfv3)#router-id 169.12.15.13
switch-C(config-router-ospfv3)#area 1 stub
```

Step d Router D OSPFv3 instance configuration:

```
switch-D(config)#ipv6 router ospf 300
switch-D(config-router-ospfv3)#router-id 169.12.15.14
```

Step e Router E OSPFv3 instance configuration:

```
switch-E(config)#ipv6 router ospf 300
switch-E(config-router-ospfv3)#router-id 169.12.15.15
```

Step f Router F OSPFv3 instance configuration:

```
switch-F(config)#ipv6 router ospf 300
switch-F(config-router-ospfv3)#router-id 169.12.15.31
```

Step 2 Configure the interfaces.

Step a Router A interfaces:

```
switch-A(config)#interface vlan 1101
switch-A(config-if-Vl1101)#ipv6 ospf 300 area 1
switch-A(config-if-Vl1101)#exit
switch-A(config)#interface vlan 1102
switch-A(config-if-Vl1102)#ipv6 ospf 300 area 1
```

Step b Router B interfaces:

```
switch-B(config)#interface vlan 1201
switch-B(config-if-Vl1201)#ipv6 ospf 300 area 1
switch-B(config-if-Vl1201)#exit
```

Step c Router C interfaces:

```
switch-C(config)#interface vlan 1301
switch-C(config-if-Vl1301)#ipv6 ospf 300 area 1
switch-C(config-if-Vl1301)#exit
switch-C(config)#interface vlan 1302
switch-C(config-if-Vl1302)#ipv6 ospf 300 area 0
```

Step d Router D interfaces:

```
switch-D(config)#interface vlan 1401
switch-D(config-if-Vl1401)#ipv6 ospf 300 area 0
switch-D(config-if-Vl1401)#exit
switch-D(config)#interface vlan 1402
switch-D(config-if-Vl1402)#ipv6 ospf 300 area 0
```

Step e Router E interfaces:

```
switch-E(config)#interface vlan 1501
switch-E(config-if-Vl1501)#ipv6 ospf 300 area 0
switch-E(config-if-Vl1501)#exit
switch-E(config)#interface vlan 1502
switch-E(config-if-Vl1502)#ipv6 ospf 300 area 0
```

Step f Router F interfaces:

```
switch-F(config)#interface vlan 1601
switch-F(config-if-Vl1601)#ipv6 ospf 300 area 0
switch-F(config-if-Vl1601)#exit
```

20.5 OSPFv3 Commands

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Mode

- ipv6 router ospf. . . . Page 875
- clear ipv6 ospf force-spf. . . . Page 863

Interface Configuration Mode

- ipv6 ospf area . . . Page 867
- ipv6 ospf cost. . . . Page 868
- ipv6 ospf dead-interval . . . Page 869
- ipv6 ospf hello-interval . . . Page 870
- ipv6 ospf network. . . . Page 871
- ipv6 ospf priority . . . Page 872
- ipv6 ospf retransmit-interval . . . Page 873
- ipv6 ospf transmit-delay . . . Page 874

Router-OSPF3 Configuration Mode

- area (OSPFv3) . . . Page 861
- area default-cost (OSPFv3) . . . Page 862
- default-metric (OSPFv3) . . . Page 864
- distance ospf intra-area (OSPFv3) . . . Page 865
- exit (OSPFv3). . . . Page 866
- log-adjacency-changes (OSPFv3). . . . Page 876
- maximum-paths (OSPFv3) . . . Page 877
- passive-interface (OSPFv3). . . . Page 878
- redistribute (OSPFv3) . . . Page 879
- router-id (OSPFv3) . . . Page 880
- shutdown (OSPFv3). . . . Page 886
- timers spf (OSPFv3) . . . Page 887

Display Commands

- show ipv6 ospf . . . Page 881
- show ipv6 ospf border-routers . . . Page 882
- show ipv6 ospf database . . . Page 883
- show ipv6 ospf interface . . . Page 884
- show ipv6 ospf neighbor. . . . Page 885

area (OSPFv3)

The **area** command configures the area type of an OSPFv3 area. All routers in an AS must specify the same area type for identically numbered areas.

The switch supports three area types:

- Normal areas: Normal areas accept intra-area, inter-area, and external routes. The backbone (area 0) is a normal area.
- Stub area: Stub areas are areas in which external routes are not advertised. To reach these external routes, a default summary route (0.0.0.0) is inserted into the stub area. Networks without external routes do not require stub areas.

Areas are ***normal*** by default; area type configuration is required only for stub NSSA areas. Area 0 is always a normal area and cannot be configured through this command.

The **no area** and **default area** commands remove the specified area from the OSPFv3 instance by deleting all **area** commands from ***running-config*** for the specified area, including the **area default-cost (OSPFv3)** command.

The **no area stub** and **default area stub** commands configure the specified area as a normal area.

Command Mode

Router-OSPF3 Configuration

Command Syntax

```
area area_id [TYPE]
no area area_id [TYPE]
default area area_id [TYPE]
```

Parameters

- *area_id* area number.

  Valid formats: integer <**1** to **4294967295**> or dotted decimal <**0.0.0.1** to **255.255.255.255**>
  Area 0 (or 0.0.0.0) is not configurable; it is always ***normal***.
  Running-config stores value in dotted decimal notation.

- ***TYPE*** area type. Values include:

  — <no parameter> area is configured as normal.
  — **stub** area is configured as a stub area.

Example

- These commands configures area 200 as a normal area and 300 as a stub area.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#area 200
switch(config-router-ospf3)#area 300 stub
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   area 0.0.0.200
   area 0.0.1.44 stub
switch(config-router-ospf3)#
```

area default-cost (OSPFv3)

The **area default-cost** command specifies the cost for the default summary routes sent into a specified area. When the **area default-cost** command is not configured for an area, the default-cost of that area is set to 10.

The **no area default-cost** and **default area default-cost** command resets the default-cost value of the specified area to 10 by removing the corresponding **area default-cost** command from ***running-config***. The **no area (OSPFv3)** command removes all area commands for the specified area from ***running-config***, including the **area default-cost** command.

Command Mode

Router-OSPF3 Configuration

Command Syntax

area *area_id* **default-cost** *def_cost*
no area *area_id* **default-cost**
default area *area_id* **default-cost**

Parameters

- *area_id* area number. <**0** to **4294967295**> or <**0.0.0.0** to **255.255.255.255**>
  Running-config stores value in dotted decimal notation.
- *def_cost* cost of the default summary route. Values range from 1 to 65535.

Example

- These commands configure a cost of 15 for default summary routes that an ABR sends into area 100.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#area 100 default 15
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   area 0.0.0.100 default-cost 15
switch(config-router-ospf3)#
```

clear ipv6 ospf force-spf

The **clear ipv6 ospf force-spf** command starts the SPF algorithm without clearing the OSPFv3 database. The SPF algorithm calculates the best path to all known destinations based on information in the link state database.

Command Mode

Global Configuration

Command Syntax

```
clear ipv6 ospf force-spf
```

Example

- This command restarts the SPF algorithm without first clearing the OSPFv3 database.

```
switch(config)#clear ipv6 ospf force-spf
switch(config)#
```

default-metric (OSPFv3)

The **default-metric** command sets default metric value for routes redistributed into the the OSPFv3 routing domain.

The **no default-metric** and **default default-metric** commands restores the default metric to its default value of 10 by removing the **default-metric** command from ***running-config***.

Command Mode

Router-OSPF3 Configuration

Command Syntax

```
default-metric def_metric
no default-metric
default default-metric
```

Parameters

- *def_metric* Cost of the default summary route. Values range from 1 to 65535. Default value is 10.

Example

- These commands configure a default metric of 30 for routes redistributed into OSPFv3.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#default-metric 30
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   default-metric 30
switch(config-router-ospf3)#
```

distance ospf intra-area (OSPFv3)

The **distance ospf intra-area** command specifies the administrative distance for routes contained in a single OSPFv3 area. Administrative distances are used when comparing dynamic routes configured through different protocols. The default administrative distance for intra-area routes is 110.

The **no distance ospf intra-area** and **default distance ospf intra-area** commands remove the **distance ospf intra-area** command from ***running-config***, returning the OSFPv3 intra-area distance setting to the default value of 110.

Command Mode

Router-OSPF3 Configuration

Command Syntax

```
distance ospf intra-area distance
no distance ospf intra-area
default distance ospf intra-area
```

Parameters

- *distance* administrative distance value. Values range from 1 to 255. Default is 110.

Example

- This command configures a distance of 90 for all OSPFv3 intra-area routes on the switch.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#distance ospf intra-area 90
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   distance ospf intra-area 90
switch(config-router-ospf3)#
```

exit (OSPFv3)

In router-ospf3 configuration mode, the **exit** command places the switch in global configuration mode. Router-ospf3 configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect ***running-config***.

Command Mode

Router-OSPF3 Configuration

Command Syntax

exit

Example

- This command exits OSPFv3 configuration mode.

```
switch(config-router-ospf3)#exit
switch(config)#
```

ipv6 ospf area

The **ipv6 ospf area** command enables OSPFv3 on the configuration mode interface and associates the specified area to the interface. Each routed interface can be associated with one OSPFv3 area; subsequent **ipv6 ospf area** commands that designate a different area on an interface replace any existing command for the interface.

OSPFv3 areas are configured in by **area (OSPFv3)** commands in router-ospf3 configuration mode

The **no ipv6 ospf area** and **default ipv6 ospf area** commands disable OSPFv3 on the configuration mode interface by removing the corresponding **ipv6 ospf area** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf process_id area area_id
no ipv6 ospf process_id [area area_id]
default ipv6 ospf process_id [area area_id]
```

Parameters

- *process_id* OSPFv3 process ID. Values range from 1 to 65535.
- *area_id* area number.

  Valid formats: integer <**0** to **4294967295**> or dotted decimal <**0.0.0.0** to **255.255.255.255**>
  Running-config stores value in dotted decimal notation.

Example

- These commands enable OSPFv3 on VLAN interface 200 and associates area 0 to the interface.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf 9 area 0
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf 9 area 0.0.0.0
switch(config-if-Vl200)#
```

ipv6 ospf cost

The **ipv6 ospf cost** command configures the OSPFv3 cost for the configuration mode interface. The OSPFv3 interface cost (or metric) typically reflects the packet transmission overhead for the interface and is inversely proportional to the interface bandwidth. The default OSPFv3 cost is 10.

The **no ipv6 ospf cost** and **default ipv6 ospf cost** commands restore the default cost of 10 for the configuration mode interface by removing the corresponding **ipv6 ospf cost** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf cost interface_cost
no ipv6 ospf cost
default ipv6 ospf cost
```

Parameters

- *interface_cost* cost assigned to the interface. Value ranges from 1 to 65535; default is 10.

Example

- This command configures a cost of 50 for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf cost 50
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf cost 50
switch(config-if-Vl200)#
```

ipv6 ospf dead-interval

The **ipv6 ospf dead-interval** command configures the OSPFv3 dead interval for the configuration mode interface.

The dead interval specifies the period that an interface waits for an OSPFv3 packet from a neighbor before it disables the adjacency under the assumption that the neighbor is down. The dead interval should be configured identically on all OSPFv3 neighbors and be longer than the hello interval of any neighbor.

The **no ipv6 ospf dead-interval** and **default ipv6 ospf dead-interval** commands restore the default dead interval of 40 seconds on the configuration mode interface by removing the corresponding **ipv6 ospf dead-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf dead-interval time
no ipv6 ospf dead-interval
default ipv6 ospf dead-interval
```

Parameters

- *time* dead interval (seconds). Value ranges from 1 to 65535; default is 40.

Example

- This command configures a dead interval of 75 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf dead-interval 75
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf dead-interval 75
switch(config-if-Vl200)#
```

ipv6 ospf hello-interval

The **ipv6 ospf hello-interval** command configures the OSPFv3 hello interval for the configuration mode interface. The hello interval defines the period between the transmission of consecutive hello packets.

Each OSPFv3 neighbor should specify the same hello interval, which should not be longer than any neighbor's dead interval.

The **no ipv6 ospf hello-interval** and **default ipv6 ospf hello-interval** commands restore the default hello interval of 10 seconds on the configuration mode interface by removing the **ipv6 ospf hello-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf hello-interval time
no ipv6 ospf hello-interval
default ipv6 ospf hello-interval
```

Parameters

- *time* hello interval (seconds). Values range from 1 to 65535; default is 10.

Example

- These commands configure a hello interval of 45 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf hello-interval 45
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf hello-interval 45
switch(config-if-Vl200)#
```

ipv6 ospf network

The **ipv6 ospf network** command sets the configuration mode interface as a point-to-point link. By default, interfaces are configured as broadcast links.

The **no ipv6 ospf network** and **default ipv6 ospf network** commands set the configuration mode interface as a broadcast link by removing the corresponding **ipv6 ospf network** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf network point-to-point
no ipv6 ospf network
default ipv6 ospf network
```

Examples

- These commands configure VLAN interface 200 as a point-to-point link.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf network point-to-point
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf network point-to-point
switch(config-if-Vl200)#
```

- This command restores Ethernet interface 10 as a broadcast link.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#no ipv6 ospf network
switch(config-if-Vl200)#show active
interface Vlan200
switch(config-if-Vl200)#
```

ipv6 ospf priority

The **ipv6 ospf priority** command configures OSPFv3 router priority for the configuration mode interface.

Router priority determines preference during designated router (DR) and backup designated router (BDR) elections. Routers with higher priority numbers have preference over other routers. The default priority is 1. Routers with a priority of zero cannot be elected as a DR or BDR.

The **no ipv6 ospf priority** and **default ipv6 ospf priority** commands restore the default priority (1) on the configuration mode interface by removing the corresponding **ipv6 ospf priority** command from *running-config.*

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf priority priority_level
no ipv6 ospf priority
default ipv6 ospf priority
```

Parameters

- *priority_level* priority level. Settings range from 0 to 255.

Example

- This command configures a router priority of 128 for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf priority 128
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf priority 128
switch(config-if-Vl200)#
```

ipv6 ospf retransmit-interval

The **ipv6 ospf retransmit-interval** command configures the link state advertisement (LSA) retransmission interval for the configuration mode interface.

Routers that send LSAs to an adjacent router expect to receive an acknowledgment from that neighbor. Routers that do not receive an acknowledgment will retransmit the LSA. The retransmission interval specifies the period between these transmissions.

The **no ipv6 ospf retransmit-interval** and **default ipv6 ospf retransmit-interval** commands restore the default retransmission interval of 5 seconds on the configuration mode interface by removing the corresponding **ipv6 ospf retransmit-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf retransmit-interval period
no ipv6 ospf retransmit-interval
default ipv6 ospf retransmit-interval
```

Parameters

- *period* retransmission interval (seconds). Value ranges from 1 to 65535; default is 5.

Example

- This command configures a retransmission interval of 25 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf retransmit-interval 25
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf retransmit-interval 25
switch(config-if-Vl200)#
```

ipv6 ospf transmit-delay

The **ipv6 ospf transmit-delay** command configures the transmission delay for OSPFv3 packets over the configuration mode interface.

The transmission delay is an estimate of the time that an interface requires to transmit a link-state update packet. OSPFv3 adds this delay to the age of outbound packets to more accurately reflect the age of the LSA when received by a neighbor.

The **no ipv6 ospf transmit-delay** and **default ipv6 ospf transmit-delay** commands restore the default transmission delay of one second on the configuration mode interface by removing the corresponding **ipv6 ospf transmit-delay** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Loopback Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ipv6 ospf transmit-delay trans
no ipv6 ospf transmit-delay
default ipv6 ospf transmit-delay
```

Parameters

- *trans* LSA transmission delay (seconds). Value ranges from 1 to 65535; default is 1.

Example

- This command configures a transmission delay of 10 seconds for VLAN 200.

```
switch(config)#interface vlan 200
switch(config-if-Vl200)#ipv6 ospf transmit-delay 10
switch(config-if-Vl200)#show active
interface Vlan200
   ipv6 ospf transmit-delay 10
switch(config-if-Vl200)#
```

ipv6 router ospf

The **ipv6 router ospf** command places the switch in router-ospf3 configuration mode and, if the switch does not contain an OSPFv3 instance, instantiates OSPFv3 and provides a process ID for the new instance. The **exit (OSPFv3)** command returns the switch to global configuration mode.

The switch supports one OSPFv3 instance, identified by its process ID. When an instance exists, this command must specify its process ID. Attempts to create additional instances will generate errors. Process IDs are local to the switch and have no effect on instances in the same AS on different routers.

The **show ipv6 ospf** command displays the process ID of the OSPFv3 instance configured on the switch.

The **no ipv6 router ospf** and **default ipv6 router ospf** command deletes the OSPFv3 instance.

These commands are available in router-ospf3 configuration modes

- **area (OSPFv3)**
- **area default-cost (OSPFv3)**
- **default-metric (OSPFv3)**
- **distance ospf intra-area (OSPFv3)**
- **exit (OSPFv3)**
- **log-adjacency-changes (OSPFv3)**
- **maximum-paths (OSPFv3)**
- **passive-interface (OSPFv3)**
- **redistribute (OSPFv3)**
- **router-id (OSPFv3)**
- **shutdown (OSPFv3)**
- **timers spf (OSPFv3)**

Command Mode

Global Configuration

Command Syntax

```
ipv6 router ospf process_id
no router ospf process_id
default router ospf process_id
```

Parameters

- *process_id* OSPFv3 process ID. Values range from 1 to 65535.

Examples

- This command creates an OSPFv3 instance with process ID 9.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#show active
ipv6 router ospf 9
switch(config-router-ospf3)#
```

- This command deletes the OSPFv3 instance.

```
switch(config)#no ipv6 router ospf 9
switch(config)#
```

log-adjacency-changes (OSPFv3)

The **log-adjacency-changes** command configures the switch to send syslog messages when it detects a neighbor has gone up or down. Log message sending is disabled by default. Valid options include:

- **log-adjacency-changes:** switch sends syslog messsages when a neighbor goes up or down (default).
- **no log-adjacency-changes** disables link state change syslog reporting.

The default option is active when ***running-config*** does not contain any form of the command. Entering the command in any form replaces the previous command state in ***running-config***. The **default log-adjacency-changes** command restores the default state by removing the **log-adjacency-changes** statement from ***running-config***.

Command Mode

Router-OSPF3 Configuration

Command Syntax

```
log-adjacency-changes
no log-adjacency-changes
default log-adjacency-changes
```

Example

- This command configures the switch to send a syslog message when a neighbor goes up or down.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#log-adjacency-changes
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   log-adjacency-changes
switch(config-router-ospf3)#
```

maximum-paths (OSPFv3)

The **maximum-paths** command controls the maximum number of parallel routes that OSPFv3 supports on the switch. The default maximum is 16 paths.

The **no maximum-paths** command restores the maximum number of parallel routes that OSPFv3 supports on the switch to the default value of 16 by removing the maximum-paths command from *running-config*.

Command Mode

Router-OSPF3 Configuration

Command Syntax

maximum-paths *paths*
no maximum-paths
default maximum-paths

Parameters

- *paths* maximum number of parallel routes. Values range from 1 to 32. Default value is 16.

Example

- This command configures the maximum number of OSPFv3 parallel paths to 12.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#maximum-paths 12
switch(config-router-ospf3)#
```

passive-interface (OSPFv3)

The **passive-interface** command disables OSPFv3 processing on an interface range. The switch neither sends OSPFv3 packets, nor processes OSPFv3 packets received on passive interfaces. The router advertises the passive interface as part of the router LSA. All interfaces are active by default.

The **no passive-interface** and **default passive-interface** commands enable OSPFv3 processing on the specified interface range by removing the corresponding **passive-interface** statements from *running-config*.

Command Mode

Router-OSPF3 Configuration

Command Syntax

```
passive-interface INTERFACE_NAME
no passive-interface INTERFACE_NAME
default passive-interface INTERFACE_NAME
```

Parameters

- *INTERFACE_NAME* interface to be configured. Options include:
  - **ethernet** *e_range* Ethernet interface list.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_range* Channel group interface list.
  - **port-channel** *p_range* Channel group interface list.
  - **vlan** *v_range* VLAN interface list.

  Valid *e_range*, *l_range*, *m_range*, *p_range*, and *v_range* formats include number, range, or comma-delimited list of numbers and ranges.

Example

- This command configures VLAN interfaces 101 through 103 as passive interfaces.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#passive-interface vlan 101-103
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   passive-interface Vlan101
   passive-interface Vlan102
   passive-interface Vlan103
switch(config-router-ospf3)#
```

redistribute (OSPFv3)

The **redistribute** command enables the advertising of all specified routes on the switch into the OSPFv3 domain as external routes. Each command enables the redistribution of one route type. ***Running-config*** allows multiple **redistribute** commands, one for each type of route to be redistributed into the OSPFv3 domain. Individual routes are not configurable for redistribution.

The **no redistribute** and **default redistribute** commands remove the corresponding **redistribute** command from ***running-config***, disabling route redistribution for the specified route type.

Command Mode

Router-OSPF3 Configuration

Command Syntax

```
redistribute ROUTE_TYPE
no redistribute ROUTE_TYPE
default redistribute ROUTE_TYPE
```

Parameters

- ***ROUTE_TYPE*** source from which routes are redistributed. Options include:
  - **connected** routes that are established when IPv6 is enabled on an interface.
  - **static** IPv6 static routes.

Example

- The **redistribute static** command starts the advertising of static routes as OSPFv3 external routes.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#redistribute static
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   redistribute connected
   redistribute static
switch(config-router-ospf3)#
```

router-id (OSPFv3)

The **router-id** command configures the router ID for an OSPFv3 instance. The router ID is a 32-bit number, expressed in dotted decimal notation, similar to an IPv4 address. This number uniquely identifies the switch within an Autonomous System. Status commands use router IDs to reference individual switches.

The switch sets the router ID to the first available alternative in the following list:

1. The **router-id** command.
2. The loopback IP address, if a loopback interface is configured on the switch.
3. The highest IP address present on the router.

The **no router-id** and **default router-id** commands remove the router ID command from ***running-config***.

Command Mode

Router-OSPF3 Configuration

Command Syntax

```
router-id identifier
no router-id
default router-id
```

Parameters

- *identifier* router ID. Value ranges from 0.0.0.0 to 255.255.255.255 (dotted decimal notation).

Example

- This command assigns 15.5.4.2 as the router ID for the OSPFv3 instance.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#router-id 15.10.1.4
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   router-id 15.10.1.4
switch(config-router-ospf3)#
```

show ipv6 ospf

The **show ipv6 ospf** command displays information about the OSPFv3 routing process that is running on the switch.

Command Mode

EXEC

Command Syntax

show ipv6 ospf

Example

- This command displays OSPFv3 routing process information..

```
switch>show ipv6 ospf
Routing Process "ospfv3 1" with ID 171.37.0.23 and Instance 0
  It is an autonomous system boundary router and is an area border router
  Hold time between two SPFs is 5
  Minimum LSA interval 5. Minimum LSA arrival 1
  It has 13 fully adjacent neighbors
  Number of areas in this router is 2. 2 normal, 0 stub, 0 nssa
  Graceful restart is enabled
    Grace period is 40
    Strict helper is enabled
  SPF algorithm last executed 00:02:59 ago
  Area 0.0.0.0
    Number of interface in this area is 8
    It is a normal area
  Area 0.0.0.2
    Number of interface in this area is 1
    It is a normal area
```

show ipv6 ospf border-routers

The **show ipv6 ospf border-routers** command displays the internal OSPFv3 routing table entries to area border routers (ABRs) and autonomous system boundary routers (ASBRs) for each OSPFv3 area.

Command Mode

EXEC

Command Syntax

show ipv6 ospf border-routers

Example

- This command displays the ABRs and ASBRs configured in the switch.

```
switch>show ipv6 ospf border-routers
Routing Process "ospf 9":
  Router 171.37.0.32 area 0.0.0.0 ASBR
  Router 171.37.0.18 area 0.0.0.0 ASBR
  Router 171.37.0.22 area 0.0.0.0 ASBR ABR
  Router 171.37.0.31 area 0.0.0.0 ASBR ABR
  Router 171.37.0.58 area 0.0.0.0 ASBR
  Router 171.37.0.37 area 0.0.0.0 ASBR
  Router 171.37.0.22 area 0.0.0.2 ASBR ABR
  Router 171.37.0.31 area 0.0.0.2 ASBR ABR

switch>
```

show ipv6 ospf database

The **show ipv6 ospf database** command displays the OSPFv3 database of link state advertisements (LSAs).

Command Mode

EXEC

Command Syntax

show ipv6 ospf database

Example

- This command displays the OSPFv3 database of link state advertisements (LSAs).

```
switch>show ipv6 ospf database
Routing Process "ospf 9":

  AS Scope LSDB

Type         Link ID     ADV Router  Age       Seq#   Checksum
 AEX         0.0.0.5    171.37.0.37   15 0x80000005   0x00be82
 AEX         0.0.0.9    171.37.0.22 1747 0x8000002b   0x00df56
 AEX         0.0.0.3    171.37.0.46  599 0x8000002d   0x00651d
<-------OUTPUT OMITTED FROM EXAMPLE-------->


  Area 0.0.0.0 LSDB

Type         Link ID     ADV Router  Age       Seq#   Checksum
 RTR         0.0.0.0    171.37.0.32  234 0x80000031   0x00585a
 NTW        0.0.0.26    171.37.0.32  271 0x80000005   0x005609
 NAP        0.0.0.26    171.37.0.32  274 0x80000005   0x00964c
<-------OUTPUT OMITTED FROM EXAMPLE-------->


  Interface vlan3911 LSDB

Type         Link ID     ADV Router  Age       Seq#   Checksum
 LNK        0.0.0.38    171.37.0.22  267 0x80000005   0x00a45a
 LNK        0.0.0.23    171.37.0.23  270 0x8000002c   0x005b7e

  Interface vlan3902 LSDB

Type         Link ID     ADV Router  Age       Seq#   Checksum
 LNK        0.0.0.17    171.37.0.11 1535 0x8000002b   0x007120
 LNK        0.0.0.37    171.37.0.22    7 0x8000002b   0x00ce23
 LNK        0.0.0.22    171.37.0.23  250 0x8000002d   0x00c350
<-------OUTPUT OMITTED FROM EXAMPLE-------->


switch>
```

show ipv6 ospf interface

The **show ipv6 ospf interface** command displays OSPFv3 information on interfaces where OSPFv3 is enabled.

Command Mode

EXEC

Command Syntax

```
show ipv6 ospf interface
```

Example

- This command displays OSPFv3 information for interfaces where OSPFv3 is enabled.

```
switch>show ipv6 ospf interface
Ethernet17 is up
  Interface Address fe80::48c:73ff:fe00:1319%Ethernet12, Area 0.0.0.0
  Network Type Broadcast, Cost 10
  Transmit Delay is 1 sec, State Backup DR, Priority 1
  Designated Router is 171.37.0.37
  Backup Designated Router is 171.37.0.23
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 1
Vlan31 is up
  Interface Address fe80::48c:73ff:fe00:1319%Vlan31, Area 0.0.0.0
  Network Type Broadcast, Cost 10
  Transmit Delay is 1 sec, State Backup DR, Priority 1
  Designated Router is 171.37.0.22
  Backup Designated Router is 171.37.0.23
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 1
Vlan32 is up
  Interface Address fe80::48c:73ff:fe00:1319%Vlan32, Area 0.0.0.0
  Network Type Broadcast, Cost 10
  Transmit Delay is 1 sec, State DR Other, Priority 1
  Designated Router is 171.37.0.11
  Backup Designated Router is 171.37.0.22
  Timer intervals configured, Hello 10, Dead 40, Retransmit 5
  Neighbor Count is 2
switch>
```

show ipv6 ospf neighbor

The **show ipv6 ospf neighbor** command displays OSPFv3 neighbor information.

Command Mode

EXEC

Command Syntax

show ipv6 ospf neighbor

Example

- This command displays the switch's neighbors.

```
switch#show ipv6 ospf neighbor
Routing Process "ospf 9":
Neighbor 171.37.0.37 priority is 1, state is Full
  In area 0.0.0.0 interface et12
  DR is 171.37.0.37 BDR is 171.37.0.23
  Options is 0
  Dead timer is due in 37 seconds
Neighbor 171.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3911
  DR is 171.37.0.22 BDR is 171.37.0.23
  Options is 0
  Dead timer is due in 31 seconds
Neighbor 171.37.0.11 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3902
  DR is 171.37.0.11 BDR is 171.37.0.22
  Options is 0
  Dead timer is due in 33 seconds
Neighbor 171.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3902
  DR is 171.37.0.11 BDR is 171.37.0.22
  Options is 0
  Dead timer is due in 31 seconds
Neighbor 171.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3923
  DR is 171.37.0.22 BDR is 171.37.0.46
  Options is 0
  Dead timer is due in 31 seconds
Neighbor 171.37.0.22 priority is 1, state is Full
  In area 0.0.0.0 interface vlan3908
  DR is 171.37.0.22 BDR is 171.37.0.21
  Options is 0
  Dead timer is due in 39 seconds
Neighbor 171.37.0.22 priority is 1, state is Full
  In area 0.0.0.2 interface vlan3992
  DR is 171.37.0.22 BDR is 171.37.0.23
  Options is 0
  Dead timer is due in 39 seconds

switch#
```

shutdown (OSPFv3)

The **shutdown** command disables OSPFv3 on the switch. Neighbor routers are notified of the shutdown and all traffic that has another path through the network will be directed to an alternate path.

OSPFv3 is disabled by default on individual interfaces and enabled through **ipv6 ospf area** commands.

The **no shutdown** and **default shutdown** commands enable the OSPFv3 instance by removing the **shutdown** statement from the OSPFv3 block in ***running-config***.

Command Mode

Router-OSPF3 Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Example

- These commands disable OSPFv3 activity on the switch.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#shutdown
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   shutdown
switch(config-router-ospf3)#
```

timers spf (OSPFv3)

The **timers spf** command configures the shortest path first (SPF) timer. The SPF timer defines the maximum interval between OSPFv3 path calculations. The default period is five seconds.

The **no timers spf** and **default timers spf** commands restore the default maximum OSPFv3 path calculation interval to five seconds by removing the **timers spf** command from ***running-config***.

Command Mode

Router-OSPF3 Configuration

Command Syntax

timers spf *spf_time*
no timers spf
default timers spf

Parameters

- *spf_time* OSPFv3 path calculation interval (seconds). Values range from 1 to 65535.

Example

- These commands set the SPF timer to twenty seconds.

```
switch(config)#ipv6 router ospf 9
switch(config-router-ospf3)#timers spf 20
switch(config-router-ospf3)#show active
ipv6 router ospf 9
   timers spf 20
switch(config-router-ospf3)#
```

Chapter 21

Border Gateway Protocol

Border Gateway Protocol (BGP) is an exterior gateway protocol (EGP) that exchanges routing information among neighboring routers in different Autonomous Systems (AS). BGP version 4 is defined by RFC 4271.

Multiprotocol BGP (BGP4+) can carry routes from multiple address families (AFI) and sub-address families (SAFI) simultaneously over a single BGP peering. Multiprotocol BGP is defined by RFC 4760.

This chapter contains the following sections.

- Section 21.1: BGP Conceptual Overview
- Section 21.2: Configuring BGP
- Section 21.3: BGP Examples
- Section 21.4: BGP Commands

Arista switches support these BGP functions:

- A single BGP instance
- Simultaneous internal (IBGP) and external (EBGP) peering
- Multiprotocol BGP

21.1 BGP Conceptual Overview

BGP is an exterior gateway protocol (EGP) that exchanges routing information among neighboring routers in different autonomous systems through TCP sessions.

BGP neighbors, or peers, are established by manual configuration commands that create a TCP session on port 179. Internal BGP (IBGP) peers operate within a single autonomous system (AS). External BGP (EBGP) peers operate between autonomous systems. Border routers are on AS boundaries and exchange information with other autonomous systems. The primary function of border routers is distributing routes. Internal routers do not distribute route updates that they receive.

BGP defines a state machine for establishing connections. BGP routers maintain a state variable for each peer-to-peer session to track connection status. The state machine consists of these states:

- **Idle**: The router initializes BGP resources, refuses inbound BGP connection attempts, initiates a TCP connection to the peer, then transitions to the ***Connect*** state.
- **Connect**: The router waits for the TCP connection to complete, then sends an OPEN message to the peer and transitions to the ***OpenSent*** state if successful. If unsuccessful, it sets the ***ConnectRetry*** timer and transitions to the ***Active*** state upon expiry.
- **Active**: The router sets the ***ConnectRetry*** timer to zero and returns to the ***Connect*** state.

- **OpenSent:** The router waits for an OPEN message from the peer. After receiving a valid message, it transitions to the ***OpenConfirm*** state.
- **OpenConfirm**: The router waits for a keepalive message from its peer. If the message is received prior to a timeout expiry, the router transitions to the ***Established*** state. If the timeout expires or an error condition exists, the router transitions to the ***Idle*** state.
- **Established**: Peers exchange UPDATE messages about routes they advertise. If an UPDATE message contains an error, the router sends a NOTIFICATION message and transitions to the ***Idle*** state.

During established BGP sessions, routers exchange UPDATE messages about the destinations to which they offer connectivity. The route description includes the destination prefix, prefix length, autonomous systems in the path, the next hop, and information that affects the acceptance policy of the receiving router. UPDATE messages also list destinations to which the router no longer offers connectivity.

BGP detects and eliminates routing loops while making routing policy decisions by using the network topology as defined by AS paths and path attributes.

Multiprotocol BGP

Multiprotocol BGP facilitates the advertisement of network routes and switch capabilities to neighbors from multiple address families over a single BGP peering. The switch supports IPv4 unicast and IPv6 unicast address families.

Neighbors negotiate to select an address family when establishing a connection. The peer session is based on this address family, which identifies the following:

- The set of network layer protocols to which the address carried in the Next Hop field must belong,
- The encoding format of the next hop address.
- The semantics of Network Layer Reachability Information (NLRI).

21.2 Configuring BGP

These sections describe basic BPG configuration steps:

- Section 21.2.1: Configuring BGP Instances
- Section 21.2.2: Configuring BGP Neighbors
- Section 21.2.3: Configuring Routes
- Section 21.2.4: Configuring Address Families
- Section 21.2.5: BGP Operational Commands

21.2.1 Configuring BGP Instances

21.2.1.1 Creating an Instance and Entering BGP Configuration Mode

The switch supports one BGP instance in a specified AS. The AS number uniquely identifies the switch to other BGP peers. BGP configuration commands apply globally to the BGP instance.

The switch must be in router-BGP configuration mode to run BGP configuration commands. The **router bgp** command places the switch in router-BGP configuration mode and creates a BGP instance if one was not previously created.

Example

- This command places the switch in router-BGP configuration mode. It also creates a BGP instance in AS 50 if an instance was not previously created.

```
switch(config)#router bgp 50
switch(config-router-bgp)#
```

When a BGP instance exists, the **router bgp** command must include its autonomous system. Any attempt to create a second instance results in an error message.

Example

- This command attempts to open a BGP instance with a different AS number from that of the existing instance. The switch displays an error and stays in global configuration mode.

```
switch(config)#router bgp 100
% BGP is already running with AS number 50
switch(config)#
```

21.2.2 Configuring BGP Neighbors

21.2.2.1 Establishing BGP Neighbors

BGP neighbors, or peers, are established by configuration commands that initiate a TCP connection. BGP supports two types of neighbors:

- Internal neighbors are in the same autonomous system.
- External neighbors are in different autonomous systems.

The **neighbor remote-as** command connects the switch with a peer.

Examples

- These commands establish an internal BGP connection with the peer at 10.1.1.14.

```
switch(config)#router bgp 50
switch(config-router-bgp)#neighbor 10.1.1.14 remote-as 50
switch(config-router-bgp)#
```

- These commands establish an external BGP connection with the peer at 20.14.1.5.

```
switch(config)#router bgp 50
switch(config-router-bgp)#neighbor 20.14.1.5 remote-as 100
switch(config-router-bgp)#
```

The **show ip bgp summary** and **show ip bgp neighbors** commands display neighbor connection status.

Example

- This command indicates the connection state with the peer at 20.14.1.5 is ***Established***. The peer is an external neighbor because it is in AS 100 and the local server is in AS 50.

```
switch>show ip bgp summary
BGP router identifier 192.168.104.2, local AS number 50
20.14.1.5        4   100    Established
switch>
```

21.2.2.2 Maintaining Neighbor Connections

BGP neighbors maintain connections by exchanging keepalive, UPDATE, and NOTIFICATION messages. Neighbors that do not receive a message from a peer within a specified period (***hold time***) close the BGP session with that peer. Hold time is typically three times the period between scheduled keepalive messages. The default keepalive period is 60 seconds; default hold time is 180 seconds.

The **timers bgp** command configures the hold time and keepalive period. A peer retains its BGP connections indefinitely when its hold time is zero.

Example

- This command configures the keepalive period of 15 seconds and hold time of 45 seconds.

```
switch(config-router-bgp)#timers bgp 15 45
switch(config-router-bgp)#
```

The **show ip bgp neighbors** command displays the hold time.

Example

- This command indicates the BGP hold time is 45 seconds.

```
switch>show ip bgp neighbors 10.100.100.2
BGP neighbor is 10.100.100.2, remote AS 100
BGP version is 4, remote router ID 192.168.104.2
Negotiated version is 4
TTL is 0
holdtime is 45                                          <= hold time
restart-time is 0
Restarting: no
Current state is Established
Updates received: 1
Updates sent: 4
Total messages received: 372
Total messages sent: 383
Last state was OpenConfirm
Last event was RecvKeepAlive
Last error code was 0
Last error subcode was 0
Local TCP address is 10.100.100.1
Local AS is 100
Local router ID is 192.168.103.1
<--------OUTPUT OMITTED FROM EXAMPLE-------->
switch>
```

21.2.2.3 Neighbor – Route Configuration

Maximum Routes

The **neighbor maximum-routes** command determines the number of BGP routes the switch accepts from a specified neighbor. The switch disables peering with the neighbor when this number is exceeded.

Example

- This command configures the switch to accept 15,000 routes from the peer at 12.1.18.24.

```
switch(config-router-bgp)#neighbor 12.1.18.24 maximum-routes 15000
switch(config-router-bgp)#
```

Route Reflection

Participating BGP routers within an AS communicate EBGP-learned routes to all of their peers, but to prevent routing loops they must not re-advertise IBGP-learned routes within the AS. To ensure that all members of the AS share the same routing information, a fully meshed network topology (in which each member router of the AS is connected to every other member) can be used, but this topology can result in high volumes of IBGP messages when it is scaled. Instead, in larger networks one or more routers can be configured as route reflectors.

A route reflector is configured to re-advertise routes learned through IBGP to a group of BGP neighbors within the AS (its clients), eliminating the need for a fully meshed topology. The **neighbor route-reflector-client** command configures the switch to act as a route reflector and configures the specified neighbor as one of its clients. The **bgp client-to-client reflection** command enables client-to-client reflection.

When using route reflectors, an AS is divided into clusters. A cluster consists of one or more route reflectors and a group of clients to which they re-advertise route information. Multiple route reflectors can be configured in the same cluster to increase redundancy and avoid a single point of failure. Each route reflector has a cluster ID. If the cluster has a single route reflector, the cluster ID is its router ID. If a cluster has multiple route reflectors, a 4-byte cluster ID is assigned to all route reflectors in the cluster. All of them must be configured with the same cluster ID so that they can recognize updates from other route reflectors in the same cluster. The **bgp cluster-id** command configures the cluster ID in a cluster with multiple route reflectors.

Example

- These commands configure the switch as a route reflector and the neighbor at 101.72.14.5 as one of its clients, and set the cluster ID to 172.22.30.101.

```
switch(config-router-bgp)#neighbor 101.72.14.5 route-reflector-client
switch(config-router-bgp)#bgp cluster-id 172.22.30.101
switch(config-router-bgp)#
```

Usually the clients of a route reflector are not interconnected, and any routes learned by a client are mirrored to other clients and re-advertised within the AS by the route reflector. If the clients of a route reflector are fully meshed, routes received from a client do not need to be mirrored to other clients. In this case, client-to-client reflection should be disabled using the **no** form of the **bgp client-to-client reflection** command.

Route Preference

The primary function of external peers is to distribute routes they learn from their peers. Internal peers receive route updates without distributing them. External peers receive route updates, then distribute them to internal and external peers.

Local preference is a metric that IBGP sessions use to select an external route. Preferred routes have the highest local preference value. UPDATE packets include this metric in the LOCAL_PREF field.

The **neighbor export-localpref** command specifies the LOCAL_PREF that the switch sends to an internal peer. The command overrides previously assigned preferences and has no effect on external peers.

Example

- This command configures the switch to enter 200 in the LOCAL_PREF field of UPDATE packets it sends to the peer at 10.1.1.45.

```
switch(config-router-bgp)#neighbor 10.1.1.45 export-localpref 200
switch(config-router-bgp)#
```

The **neighbor import-localpref** command assigns a local preference to routes received through UPDATE packets from an external peer. This command has no affect when the neighbor is an internal peer.

Example

- This command configures the switch to assign the local preference of 50 for routes advertised from the peer at 14.4.1.30.

```
switch(config-router-bgp)#neighbor 14.4.1.30 import-localpref 50
switch(config-router-bgp)#
```

The **show ip bgp** command displays the LOCAL_PREF value for all listed routes.

Example

- This command indicates the route to network 10.10.20.0/24 has a local preference of 400.

```
switch#show ip bgp
Route status codes: s - suppressed, * - valid, > - active

    Network         Next Hop     R Metric  LocPref Path
* > 10.10.20.0/24  10.10.10.1   u 0       400     (100) IGP (Id 4) Rt-ID: 19.16.1.1
```

21.2.2.4 Filtering Routes

Route Maps

Route maps are used in BGP to directly filter IPv4 unicast routes. The **neighbor route-map (BGP)** command applies a route map to inbound or outbound BGP routes.

Refer to Section 13.2.2: Route Maps for more information about route maps.

AS Path Access Lists

An AS path access list is a named list of permit and deny statements which use regular expressions to filter BGP routes based on their AS path attribute. AS path access lists are created using the **ip as-path access-list** command, and are applied using a route map ***match*** clause with the name of the access list as a parameter.

Example

These commands create an AS path access list identifying routes which pass through AS 3, create a route map which references the access list and assigns the routes it filters to community 300, and apply the route map to the neighbor at 101.72.14.5 to assign a community value of 300 to inbound routes received from that neighbor.

Step 1 Create the AS path access list.

```
switch(config)#ip as-path access-list as_list3 permit _3_
```

Step 2 Create a route map that matches the AS path access list and sets the community value.

```
switch(config)#route-map MAP_3 permit
switch(config-route-map-MAP_3)#match as-path as_list3
switch(config-route-map-MAP_3)#set community 300
switch(config-route-map-MAP_3)#exit
```

Step 3 Apply the route map to the neighbor.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 101.72.14.5 route-map MAP_3 in
switch(config-router-bgp)#
```

BGP Communities

A BGP community is a group of subnet address prefixes that share a common identifying attribute. Communities simplify routing policies by consolidating IP network spaces into logical entities that BGP speakers can address to accept, prefer, and distribute routing information.

The BGP community attribute is a 32 bit value formatted as follows:

- an integer between 0 and 4294967040.
- AA:NN, where AA specifies an Autonomous System number (0-65535) and NN specifies a community number (0-65535) within the AS.

These four community attribute values, and the associated BGP speaker actions, are predefined:

- **no-export**: speaker does not advertise the routes beyond the BGP domain.
- **no-advertise**: speaker does not advertise the routes to any BGP peers.
- **local-as**: speaker does not advertise route to any external peers.
- **internet**: speaker advertises the route to Internet community. By default, this includes all prefixes.

Community values are assigned to a set of subnet prefixes through route map **set** commands. Route map **match** commands subsequently use community values to filter routes. The switch uses the following **ip community-list** commands to filter community routes into a BGP domain:

- **ip community-list standard**
- **ip community-list expanded**
- **ip extcommunity-list standard**
- **ip extcommunity-list expanded**

  Standard community lists refer to route maps by name or number. Expanded community lists reference route maps through regular expressions.

  Example

  These commands assign two network subnets to a prefix list, assign a community number to the prefix list members, then utilize that community in an **ip community-list** command to permit the routes into the BGP domain.

  Step 1 Compose the IP prefix list.

  ```
  switch(config)#ip prefix-list PL_1 permit 10.1.2.5/24
  switch(config)#ip prefix-list PL_1 permit 15.2.5.1/28
  switch(config)#
  ```

  Step 2 Create a route map that matches the IP prefix list and sets the community value.

  ```
  switch(config)#route-map MAP_1 permit
  switch(config-route-map-MAP_1)#match ip address prefix-list PL_1
  switch(config-route-map-MAP_1)#set community 500
  switch(config-route-map-MAP_1)#exit
  ```

Step 3 Create a community list that references the community.

```
switch(config)#ip community-list standard CL_1 permit 500
switch(config)#
```

BGP extended communities configure, filter, and identify routes for virtual routing, forwarding instances (VRFs), and Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs).

Extended community clauses provide route target and site of origin parameter options:

- **route targets** (**rt**): This attribute identifies a set of sites and VRFs that may receive routes tagged with the configured route target. Configuring this attribute with a route allows that route to be placed in per-site forwarding tables that route traffic received from corresponding sites.
- **site of origin** (**soo**): This attribute identifies the site from where the Provider Edge (PE) router learns the route. All routes learned from a specific site have the same SOO extended community attribute, whether a site is connected to a single or multiple PE routers. This attribute prevents routing loops resulting from multihomed sites. The SOO attribute is configured on the interface and propagated into a BGP domain by redistribution. The SOO is applied to routes learned from VRFs.

21.2.3 Configuring Routes

21.2.3.1 Advertising Routes

A BGP neighbor advertises routes it can reach through UPDATE packets. The **network (BGP)** command specifies a prefix that the switch advertises as a route originating from its AS.

The configuration clears the host portion of addresses entered in **network** commands. For example, 192.0.2.4/24 is stored as 192.0.2.0/24.

Example

- This command configures the switch to advertise the 14.5.8.0/24 network.

```
switch(config-router-bgp)#network 14.5.8.0/24
switch(config-router-bgp)#
```

21.2.3.2 BGP Route Aggregation

Aggregation combines the characteristics of multiple routes into a single route for advertising by the BGP speaker. Aggregation can reduce the amount of information that a BGP speaker is required to store and transmit when advertising routes to other BGP speakers. Aggregation options affect attributes associated with the aggregated route and the advertisement of the contributor routes that comprise the aggregate route. Contributor routes with different type codes cannot be aggregated.

Aggregate routes are created with the **aggregate-address** command. BGP speakers display aggregate routes that they create as null routes. Aggregate routes are advertised into the BGP autonomous system with the **redistribute (BGP)** command. BGP neighbors display inbound aggregate routes as normal BGP routes. Null routes are displayed with the **show ip route** command; normal BGP routes are displayed with the **show ip bgp** and **show ip route** commands.

Aggregation Options

The **aggregate-address** command provides the following aggregate route options:

- AS_PATH attribute inclusion: the ***as-set*** option controls the aggregate route's AS_PATH and ATOMIC_AGGREGATE attribute contents. AS_PATH identifies the autonomous systems through which UPDATE message routing information passes. ATOMIC_AGGREGATE indicates that the route is an aggregate or summary of more specific routes.

When the command includes ***as-set***, the aggregate route's AS_SET attribute contains contributor route path elements.

When the command does not include ***as-set***, the aggregate route's ATOMIC_AGGREGATE attribute is set and AS_PATH attribute does not include data from contributing routes.

- Attribute inclusion: The ***attribute-map*** option assigns attributes contained in set commands in a specified route map's permit clauses to the aggregated route.
- Route suppression: The ***summary-only*** option suppresses the advertisement of the contributor routes that comprise the aggregate.

Examples

- These commands create an aggregate route (168.16.48.0/20) from four contributor routes (168.16.48.0/23, 168.16.50.0/23, 168.16.52.0/23, and 168.16.54.0/23). The aggregate route includes the AS_PATH information from the contributor routes.

```
switch(config)#router bgp 1
switch(config-router-bgp)#aggregate-address 168.16.48.0/20 as-set
switch(config-router-bgp)#exit
switch(config)#
```

- These commands redistribute the aggregate route into the BGP domain. The switch begins advertising the aggregate route after running these commands.

```
switch(config)#router bgp 1
switch(config-router-bgp)#redistribute aggregate
switch(config-router-bgp)#exit
switch(config)#
```

- These commands create an aggregate route and use a route map to add a local-preference attribute to the route.

```
switch(config)#route-map map1 permit 10
switch(config-route-map-map1)#set local-preference 40
switch(config-route-map-map1)#exit
switch(config)#router bgp 1
switch(config-router-bgp)#aggregate-address 168.16.48.0/20 attribute-map map1
switch(config-router-bgp)#exit
switch(config)#
```

21.2.4 Configuring Address Families

The switch determines the network prefixes that peering sessions advertise and the BGP neighbor addresses that receive advertisements through address family activity configuration.

An address family is a data structure that defines route advertising status to BGP neighbor addresses. Each BGP neighbor address is assigned an activity level for each address family on the switch. The switch sends capability and network prefix advertisements to neighbor addresses that are active within specified address families:

- IPv4 address family: switch advertises IPv4 capability and network commands with IPv4 prefixes to neighbor addresses configured as ***IPv4 address family active***.
- IPv6 address family: switch advertises IPv6 capability and network commands with IPv6 prefixes to neighbor addresses configured as ***IPv6 address family active***.

Limitation The switch does not support IPv4 neighbor addresses as ***IPv6 address family active***.

21.2.4.1 Neighbor Address Family Configuration

Address family activity levels for neighbor addresses is configured through **bgp default** and **neighbor activate** commands.

- The **bgp default** command specifies the default activity level of BGP neighbor addresses for a specified address family.
- The **neighbor activate** command specifies deviations from default address family activity level for a specified BGP neighbor address.

Default neighbor activation

The **bgp default** command configures the default address family activity level of all configured BGP neighbor addresses. The switch advertises the following to ***address family active*** addresses:

- IPv4 address family active: IPv4 capability and all network advertisements with IPv4 prefixes.
- IPv6 address family active: IPv6 capability and all network advertisements with IPv6 prefixes.

These commands configure default address family activity levels for configured BGP neighbor addresses:

- **bgp default ipv4-unicast** all BGP neighbor addresses are IPv4 address family active.
- **no bgp default ipv4-unicast** no BGP neighbor addresses are IPv4 address family active.
- **bgp default ipv6-unicast** all BGP neighbor addresses are IPv6 address family active.
- **no bgp default ipv6-unicast** no BGP neighbor addresses are IPv6 address family active.

The *default-default address family* activity state is the address family activity level of configured BGP addresses when ***running-config*** does not contain **bgp default** commands. The default address family default activity level varies by address family.

- **IPv4 address family** all BGP addresses are IPv4 address family active.
- **IPv6 address family** no BGP addresses are IPv6 address family active.

Activating Individual Neighbor Addresses

The **address-family** command places the switch in address family mode to configure the address family activity level of individual BGP neighbor addresses. The switch supports these address families:

- ipv4-unicast
- ipv6-unicast

Running-config displays address family commands in sub-blocks of the BGP configuration. The **neighbor activate** command is available in each address family configuration mode and defines the configuration mode address family activity level of a specified configured BGP neighbor address. Addresses are assigned one of the following states by the activate command:

- **neighbor activate** configures the address as active in the configuration mode address family.
- **no neighbor activate** configures the address as not active in the configuration mode address family.

The switch sends the following announcements to addresses that are active in an address family:

- IPv4 address family: IPv4 capability and all network routes with IPv4 prefixes.
- IPv6 address family: IPv6 capability and all network routes with IPv6 prefixes.

The **neighbor route-map (BGP)** command applies a route map to inbound or outbound BGP routes. In address-family mode, the route map is applied to routes corresponding to the configuration mode address family. When a route map is applied to outbound routes, the switch advertises only routes matching at least one section of the route map. One outbound and one inbound route map can be applied to a neighbor for each address family. Applying a route map to a route replaces the previous corresponding route map assignment.

Network Route Advertising in Address Families

The **network (BGP)** command specifies a network for advertisement through UPDATE packets to BGP peers. The command is available in Router-BGP and Router-BGP-Address-Family configuration modes; the mode in which the command is issued does not affect the command's execution.

- Commands with an IPv4 address are advertised to peers that are IPv4 address family-active.
- Commands with an IPv6 address are advertised to peers that are IPv6 address family-active.

Examples

- These commands instantiate BGP, configure three neighbors, and configure 2 network routes.

  The default activity level for IPv4 and IPv6 address families is set to the default; all neighbor addresses are IPv4 address family active and IPv6 address family not active. IPv4 capability and network routes with IPv4 prefixes are advertised to all neighbor IPv4 addresses.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 172.21.14.8 remote-as 15
switch(config-router-bgp)#neighbor 172.23.18.6 remote-as 16
switch(config-router-bgp)#neighbor fd7a:2433:8c01::1 remote-as 16
switch(config-router-bgp)#network 171.18.23.9/24
switch(config-router-bgp)#network fd7b:3822:de29::/64
switch(config-router-bgp)#
```

- These commands instantiate BGP on the switch, sets IPv4 default activity level to ***not active***, sets the IPv6 default activity level to ***active***, configures three neighbors addresses and two network route prefixes.

  IPv6 capability and network routes with IPv6 prefixes are advertised to all neighbor addresses.

```
switch(config)#router bgp 10
switch(config-router-bgp)#bgp default ipv6-unicast
switch(config-router-bgp)#no bgp default ipv4-unicast
switch(config-router-bgp)#neighbor 172.21.14.8 remote-as 15
switch(config-router-bgp)#neighbor 172.23.18.6 remote-as 16
switch(config-router-bgp)#neighbor fd7a:2433:8c01::1 remote-as 16
switch(config-router-bgp)#network 171.18.23.9/24
switch(config-router-bgp)#network fd7b:3822:de29::/64
switch(config-router-bgp)#
```

- These commands configure threee neighbors, two network routes, the default activity level for each address family as not active, and specifies neighbor addresses for each address family that are active.

```
switch(config)#router bgp 11
switch(config-router-bgp)#neighbor 172.21.14.8 remote-as 15
switch(config-router-bgp)#neighbor 172.23.18.6 remote-as 16
switch(config-router-bgp)#neighbor fd7a:2433:8c01::1 remote-as 16
switch(config-router-bgp)#network 171.18.23.9/24
switch(config-router-bgp)#network fd7b:3822:de29::/64
switch(config-router-bgp)#no bgp default ipv4-unicast
switch(config-router-bgp)#no bgp default ipv6-unicast
switch(config-router-bgp)#address-family ipv4
switch(config-router-bgp-af)#neighbor 172.21.14.8 activate
switch(config-router-bgp-af)#neighbor 172.23.18.6 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#address-family ipv6
switch(config-router-bgp-af)#neighbor fd7a:2433:8c01::1 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#
```

21.2.5 BGP Operational Commands

21.2.5.1 Shutdown

The **shutdown (BGP)** command disables BGP operations without disrupting the BGP configuration. The **no router bgp** command disables BGP and removes the BGP configuration.

The **no shutdown** command resumes BGP activity.

Examples

- This command disables BGP activity on the switch.

```
switch(config-router-bgp)#shutdown
switch(config-router-bgp)#
```

- This command resumes BGP activity on the switch.

```
switch(config-router-bgp)#no shutdown
switch(config-router-bgp)#
```

21.2.5.2 Clearing the Routing Table and Resetting BGP Sessions

Changes to a route map do not take effect until the BGP process is forced to recognize the changes. The **clear ip bgp** command clears all BGP learned routes from the routing table, reads routes from designated peers, and sends routes required by those peers. Routes that are read or sent are processed through any modified route map or AS-path access list.

The **clear ip bgp *** command clears the BGP sessions with all BGP peers. To reset the session with a specific peer, enter the peer's IP address in place of the asterisk.

Example

- This command removes all BGP learned routes from the routing table.

```
switch#clear ip bgp
switch#
```

21.3 BGP Examples

This section describes the commands required to configure an IBGP and an EBGP topology

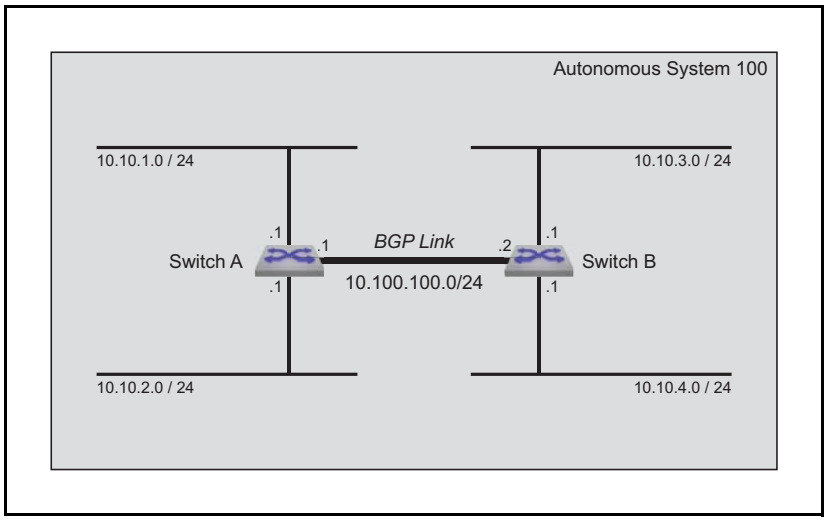
21.3.1 Example 1

Example 1 features an internal BGP link that connects peers in AS 100.

21.3.1.1 Diagram

Figure 21-1 displays BGP Example 1. The BGP link establishes IBGP neighbors in AS 100. Each switch advertises two subnets. In UPDATE packets sent by Switch A, the LOCAL_PREF field is 150. In UPDATE packets sent by Switch B, the LOCAL_PREF field is 75.

Figure 21-1 BGP Example 1



21.3.1.2 Code

This code configures the Example 1 BGP instance on both switches.

Step 1 Configure the neighbor addresses.

Step a Specify the neighbor to Switch A.

```
switchA(config)#router bgp 100
switchA(config-router-bgp)#neighbor 10.100.100.2 remote-as 100
```

Step b Specify the neighbor to Switch B.

```
switchB(config)#router bgp 100
switchB(config-router-bgp)#neighbor 10.100.100.1 remote-as 100
```

Step 2 Configure the routes to be advertised

Step a Advertise Switch A's routes.

```
switchA(config-router-bgp)#network 10.10.1.0/24
switchA(config-router-bgp)#network 10.10.2.0/24
```

Step b Advertise Switch B's routes.

```
switchB(config-router-bgp)#network 10.10.3.0/24
switchB(config-router-bgp)#network 10.10.4.0/24
```

Step 3 Configure the LOCAL_PREF.

```
switchA(config-router-bgp)#neighbor 10.100.100.2 export-localpref 150
switchB(config-router-bgp)#neighbor 10.100.100.1 export-localpref 75
```

Step 4 Modify the hold time and keepalive interval.

```
switchA(config-router-bgp)#timer bgp 30 90
switchB(config-router-bgp)#timer bgp 30 90
```

21.3.2 Example 2

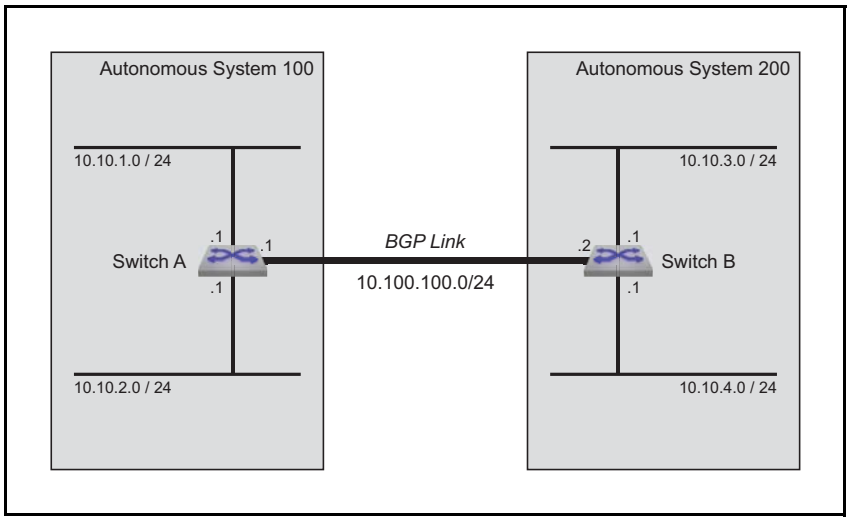
Example 2 creates an external BGP link that connects routers in AS 100 and AS 200.

21.3.2.1 Diagram

Figure 21-2 displays BGP Example 2. The BGP link connects a switch in AS 100 to a switch in AS 200. Each switch advertises two subnets.

Switch A assigns a local preference of 150 to networks advertised by Switch B. Switch B assigns a local preference of 75 to networks advertised by Switch A.

Figure 21-2 BGP Example 2



21.3.2.2 Code

This code configures the Example 2 BGP instance on both switches.

Step 1 Configure the neighbor addresses.

Step a Specify the neighbor to Switch A.

```
switchA(config)#router bgp 100
switchA(config-router-bgp)#neighbor 10.100.100.2 remote-as 200
```

Step b Specify the neighbor to Switch B.

```
switchB(config)#router bgp 200
switchB(config-router-bgp)#neighbor 10.100.100.1 remote-as 100
```

Step 2 Configure the routes to be advertised

Step a Advertise Switch A's routes.

```
switchA(config-router-bgp)#network 10.10.1.0/24
switchA(config-router-bgp)#network 10.10.2.0/24
```

Step b Advertise Switch B's routes.

```
switchB(config-router-bgp)#network 10.10.3.0/24
switchB(config-router-bgp)#network 10.10.4.0/24
```

Step 3 Assign local preference values to routes received from their respective peers.

```
switchA(config-router-bgp)#neighbor 10.100.100.2 import-localpref 150
switchB(config-router-bgp)#neighbor 10.100.100.2 import-localpref 75
```

Step 4 Modify the hold timer and keepalive interval.

```
switchA(config-router-bgp)#timer bgp 30 90
switchB(config-router-bgp)#timer bgp 30 90
```

21.4 BGP Commands

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Commands

- router bgp . . . . . Page 955
- ip as-path access-list . . . . . Page 919
- ip community-list expanded . . . . . Page 920
- ip community-list standard . . . . . Page 921
- ip extcommunity-list expanded . . . . . Page 922
- ip extcommunity-list standard . . . . . Page 923

Router-BGP Configuration Mode (Includes Address-Family Mode)

- address-family . . . . . Page 906
- aggregate-address . . . . . Page 907
- bgp client-to-client reflection . . . . . Page 909
- bgp cluster-id . . . . . Page 910
- bgp default . . . . . Page 911
- bgp listen limit . . . . . Page 913
- bgp listen range . . . . . Page 914
- bgp log-neighbor-changes . . . . . Page 915
- distance bgp . . . . . Page 918
- maximum paths (BGP) . . . . . Page 924
- no neighbor . . . . . Page 952
- neighbor activate . . . . . Page 925
- neighbor allowas-in . . . . . Page 926
- neighbor description . . . . . Page 927
- neighbor ebgp-multihop . . . . . Page 928
- neighbor export-localpref . . . . . Page 929
- neighbor import-localpref . . . . . Page 930
- neighbor local-as . . . . . Page 931
- neighbor local-v6-addr . . . . . Page 932
- neighbor maximum-routes . . . . . Page 933
- neighbor next-hop-peer . . . . . Page 934
- neighbor next-hop-self . . . . . Page 935
- neighbor out-delay . . . . . Page 936
- neighbor password . . . . . Page 937
- neighbor peer-group (create) . . . . . Page 938
- neighbor peer-group (neighbor assignment) . . . . . Page 940
- neighbor remote-as . . . . . Page 941
- neighbor remove-private-as . . . . . Page 942
- neighbor route-map (BGP) . . . . . Page 943
- neighbor route-reflector-client . . . . . Page 944
- neighbor send-community . . . . . Page 945
- neighbor shutdown . . . . . Page 946
- neighbor soft-reconfiguration . . . . . Page 947
- neighbor timers . . . . . Page 948
- neighbor update-source . . . . . Page 949
- neighbor weight . . . . . Page 950
- network (BGP) . . . . . Page 951
- redistribute (BGP) . . . . . Page 953
- router-id (BGP) . . . . . Page 954
- shutdown (BGP) . . . . . Page 973
- timers bgp . . . . . Page 974

Clear Commands – Privileged EXEC Mode

- clear ip bgp . . . . . Page 916
- clear ipv6 bgp . . . . . Page 917

Display Commands – EXEC Mode

- show ip as-path access-list . . . . . Page 956
- show ip bgp . . . . . Page 957
- show ip bgp community . . . . . Page 958
- show ip bgp neighbors . . . . . Page 959
- show ip bgp neighbors (route type) . . . . . Page 960
- show ip bgp neighbors (route-type) community . . . . . Page 961
- show ip bgp paths . . . . . Page 962
- show ip bgp peer-group . . . . . Page 963
- show ip bgp summary . . . . . Page 964
- show ip community-list . . . . . Page 965
- show ip extcommunity-list . . . . . Page 966
- show ipv6 bgp . . . . . Page 967
- show ipv6 bgp community . . . . . Page 968
- show ipv6 bgp neighbors . . . . . Page 969
- show ipv6 bgp neighbors (route type) . . . . . Page 970
- show ipv6 bgp neighbors (route type) community . . . . . Page 971
- show ipv6 bgp summary . . . . . Page 972

address-family

The **address-family** command places the switch in address-family configuration mode to configure the address family setting of addresses configured as BGP neighbors. Address-family configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

The switch supports these address families:

- ipv4-unicast
- ipv6-unicast

Running-config display address family commands in sub-blocks of the BGP configuration. The following commands are available in address family configuration mode:

- **neighbor activate** configures the address as active in the configuration mode address family.
- **no neighbor activate** configures the address as not active in the configuration mode address family.
- **neighbor route-map (BGP)** applies a route map to the specified BGP route.
- **network (BGP)** specifies a network for advertisement through UPDATE packets to BGP peers.

Address-family configuration mode is not a group change mode; the configuration is changed immediately after commands are executed. The **exit** command does not affect the configuration.

The **no address-family** and **default address-family** commands delete the specified address-family from ***running-config*** by removing all commands previously configured in the corresponding address-family mode.

The **exit** command returns the switch to router-BGP configuration mode.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp ADDRESS_TYPE
no bgp ADDRESS_TYPE
default bgp ADDRESS_TYPE
```

Parameters

- *ADDRESS_FAMILY* Address family affected by subsequent commands. Options include:
  - **ipv4** IPv4 unicast
  - **ipv6** IPv6 unicast

Example

- These commands enter address family mode for IPv6-unicast, insert a command, then exits the mode:

```
switch(config)#router bgp 1
switch(config-router-bgp)#address-family ipv6
switch(config-router-bgp-af)#neighbor 172.10.1.1 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#
```

aggregate-address

The **aggregate-address** command creates an aggregate route in the Border Gateway Protocol (BGP) database. Aggregate routes combine the characteristics of multiple routes into a single route that the switch advertises. Aggregation can reduce the amount of information that a BGP speaker is required to store and transmit when advertising routes to other BGP speakers. Aggregate routes are advertised only after they are redistributed.

Command options affect attributes associated with the aggregated route and the advertisement of the contributor routes that comprise the aggregate route. Contributor routes with different type codes cannot be aggregated.

Command options affect the following aggregate routing attributes:

- AS_PATH attribute inclusion: AS_PATH is an attribute that identifies the autonomous systems through which UPDATE messages carry routing information. When the command contains the **as-set** option, the aggregate route includes AS_PATH information from the contributor routes as AS_SET attributes.

  When the command does not include the **as-set** option, the ATOMIC_AGGREGATE attribute is set on the aggregate route. The aggregate route does not contain AS_PATH information.
- Attribute inclusion: The **attribute-map** option assigns attributes contained in the **set** commands of permit clauses of the specified route map to the aggregated route.
- Route suppression: The **summary-only** option suppresses the advertisement of the contributor routes that comprise the aggregate.

The **no aggregate-address** and **default aggregate-address** commands remove the corresponding **aggregate-address** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
aggregate-address AGGREGATE_NET [AS_SET] [SUMMARY] [ATTRIBUTE_MAP]
no aggregate-address AGGREGATE_NET
default aggregate-address AGGREGATE_NET
```

Parameters

- ***AGGREGATE_NET*** aggregate route IP address. Options include:
  - *netv4_addr* IPv4 subnet address (CIDR or address-mask notation).
  - *netv6_addr* neighbor's IPv6 prefix (CIDR notation).
- ***AS_SET*** controls AS_PATH attribute values associated with aggregate route. Options include:
  - <no parameter> ATOMIC_AGGREGATE attribute is set. Route contains no AS_PATH data.
  - **as-set** route includes AS_PATH information from contributor routes as AS_SET attributes.
- ***SUMMARY*** controls advertisement of contributor routes. Options include:
  - <no parameter> contributor and aggregate routes are advertised.
  - **summary-only** contributor routes are not advertised.
- ***ATTRIBUTE_MAP*** controls attribute assignments to the aggregate route. Options include:
  - <no parameter> attribute values are not assigned to route.
  - **attribute-map** *map_name* assigns attribute values in set commands of the map's permit clauses. Deny clauses and match commands in permit clauses are ignored.

Examples

- These commands create an aggregate route (168.16.48.0/20) from the contributor routes 168.16.48.0/23, 168.16.50.0/23,168.16.52.0/23, and 168.16.54.0/23. The aggregate route includes the AS_PATH information from the contributor routes.

```
switch(config)#router bgp 1
switch(config-router-bgp)#aggregate-address 168.16.48.0/20 as-set
switch(config-router-bgp)#exit
switch(config)#
```

- These commands redistribute the aggregate route into the BGP domain. The switch begins advertising the aggregate route after this command is configured.

```
switch(config)#router bgp 1
switch(config-router-bgp)#redistribute aggregate
switch(config-router-bgp)#exit
switch(config)#
```

- These commands create an aggregate route and use a route map to add a local-preference attribute to the route.

```
switch(config)#route-map map1 permit 10
switch(config-route-map-map1)#set community 45
switch(config-route-map-map1)#exit
switch(config)#router bgp 1
switch(config-router-bgp)#aggregate-address 168.16.48.0/20 attribute-map map1
switch1(config-router-bgp)#exit
switch(config)#
```

bgp client-to-client reflection

By default, routes received from a route reflector client and selected as best routes are propagated to all BGP peers, including other route reflector clients. If the clients are fully meshed, however, routes received from a client do not need to be mirrored to other clients. In this case, client-to-client reflection should be disabled.

The **no bgp client-to-client reflection** command disables client-to-client reflection.

The **bgp client-to-client reflection** and **default bgp client-to-client reflection** commands restore the default behavior by removing the **no bgp client-to-client reflection** command from ***running-config*.** Only the **no** form of this command is visible in ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp client-to-client reflection
no bgp client-to-client reflection
default bgp client-to-client reflection
```

Example

- This command disables client-to-client reflection on the switch.

```
switch(config)#router bgp 1
switch(config-router-bgp)#no bgp client-to-client reflection
switch(config-router-bgp)#
```

bgp cluster-id

When using route reflectors, an AS is divided into clusters. A cluster consists of one or more route reflectors and a group of clients to which they re-advertise route information. Multiple route reflectors can be configured in the same cluster to increase redundancy and avoid a single point of failure. Each route reflector has a cluster ID. If the cluster has a single route reflector, the cluster ID is its router ID. If a cluster has multiple route reflectors, a 4-byte cluster ID must be assigned to all route reflectors in the cluster. All of them must be configured with the same cluster ID so that they can recognize updates from other route reflectors in the same cluster.

The **bgp cluster-id** command configures the cluster ID in a cluster with multiple route reflectors.

The **no bgp cluster-id** and **default bgp cluster-id** commands remove the cluster ID by removing the corresponding **bgp cluster-id** command from ***running-config***. Do not remove the cluster ID if there are multiple route reflectors in the cluster.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp cluster-id ID_NUM
no bgp cluster-id
default bgp cluster-id
```

Parameters

- ***ID_NUM*** cluster ID shared by all route reflectors in the cluster (32-bit dotted-decimal notation). Options include:
  - — ***0.0.0.1*** to ***255.255.255.255*** Valid cluster ID number
  - — ***0.0.0.0*** removes the cluster-ID from the switch. Equivalent to **no bgp cluster-id** command.

Example

- This command sets the cluster ID for the switch to 172.22.30.101.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp cluster-id 172.22.30.101
switch(config-router-bgp)#
```

bgp default

The **bgp default** command configures the default address family activation level of all addresses configured as BGP neighbors. The switch sends the following announcements to addresses active in an address family:

- IPv4 address family: IPv4 capability and all network advertisements with IPv4 prefixes.
- IPv6 address family: IPv6 capability and all network advertisements with IPv6 prefixes.

The following commands configure default address family activation levels for addresses configured as BGP neighbors:

- **bgp default ipv4-unicast** all addresses are IPv4 address family active.
- **no bgp default ipv4-unicast** all addresses are not IPv4 address family active.
- **bgp default ipv6-unicast** all addresses are IPv6 address family active
- **no bgp default ipv6-unicast** all addresses are not IPv6 address family active.

The activation state of an individual BGP neighbor address is configured by the **neighbor activate** commands. The **neighbor activate** command overrides the address's default activation state for the address family configuration mode in which the command is issued:

- **neighbor activate**: the specified address is active.
- **no neighbor activate**: the specified address is not active.

The *default-default address family* activation state defines address family activation level of all addresses configured as BGP neighbors when ***running-config*** does not contain any **bgp default** commands. The default state of the BGP default activation level varies by address family.

- **IPv4 address family** all BGP addresses are IPv4 address family active.
- **IPv6 address family** all BGP addresses are not IPv6 address family active.

The **default bgp default** command restores the default-default activation setting for BGP neighbor addresses in the specified address family:

- **default bgp ipv4-unicast** is equivalent to **bgp ipv4-unicast**
- **default bgp ipv6-unicast** is equivalent to **no bgp ipv6-unicast**

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp default ADDRESS_FAMILY
no bgp default ADDRESS_FAMILY
default bgp default ADDRESS_FAMILY
```

Parameters

- *ADDRESS_FAMILY* BGP address family. Options include:
  - **ipv4-unicast** IPv4-unicast peering sessions.
  - **ipv6-unicast** IPv6-unicast peering sessions.

Limitations

The switch supports the advertisement of networks with IPv6 prefixes to IPv4 transport neighbors. The switch does not support the advertisement of networks with IPv4 prefixes to IPv6 transport neighbors.

Example

- These commands configure the switch to configure all BGP neighbor addresses as IPv4 address family active and IPv6 address family active.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp default ipv4-unicast
switch(config-router-bgp)#bgp default ipv6-unicast
switch(config-router-bgp)#show active
router bgp 65533
   bgp log-neighbor-changes
   distance bgp 20 200 200
   neighbor 171.23.254.2 remote-as 65533
   neighbor 171.41.254.78 remote-as 65534
   neighbor fd7c:629f:52a4:fe01::2 remote-as 65533
   neighbor fd7a:629f:52a4:fe4c::1 out-delay 10
switch(config-router-bgp)#
```

The show active command does not display the **bgp default ipv4-unicast** command because it is the default setting for IPv4 peering sessions.

bgp listen limit

The **bgp listen limit** command limits the number of dynamic BGP peers allowed on the switch.

The **no bgp listen limit** and **default bgp listen limit** commands restore the default limit of dynamic BGP peers by removing the **bgp listen limit** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp listen limit maximum
no bgp listen limit
default bgp listen limit
```

Parameters

- *maximum* the maximum number of dynamic BGP peers to be allowed on the switch. Values range from 1 to 1000; default value is 100.

Example

- This command sets the maximum number of dynamic BGP peers allowed on the switch to 200.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp listen limit 200
switch(config-router-bgp)#
```

bgp listen range

The **bgp listen range** command identifies a range of IPv4 addresses from which the switch will accept incoming dynamic BGP peering requests, and creates the named peer group to which those peers will belong. Once a peer group is created with this command, the following ***neighbor*** commands can use the peer group name as a parameter:

- **neighbor ebgp-multihop**
- **neighbor import-localpref**
- **neighbor maximum-routes**
- **neighbor route-map (BGP)**
- **neighbor timers**
- **neighbor update-source**

The **no bgp listen range** and **default bgp listen range** commands remove the peer group by deleting the corresponding command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp listen range NET_ADDRESS peer-group group_name remote-as as_number
no bgp listen range NET_ADDRESS peer-group group_name
default bgp listen range NET_ADDRESS peer-group group_name
```

Parameters

- ***NET_ADDRESS*** IPv4 address range. Entry options include:
  - *IPv4 subnet* IPv4 subnet (CIDR notation).
  - *IPv4_address* **mask** *subnet* IPv4 subnet (dotted decimal notation).
- *group_name* name of the peer group.
- *as_number* the autonomous system to which the peer group belongs.

Example

- This command creates a peer group called “brazil” in AS 5 which accepts dynamic peering requests from the 201.6.6.0/24 subnet.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp listen range 201.6.6.0/24 peer-group brazil
remote-as 5
switch(config-router-bgp)#
```

bgp log-neighbor-changes

The **bgp log-neighbor-changes** command configures the switch to generate a log message when a BGP peer enters or exits the Established state. This is the default behavior.

The **no bgp log-neighbor-changes** command disables the generation of these log messages. The **default bgp log-neighbor-changes** command enables the generation of these log messages.

Command Mode

Router-BGP Configuration

Command Syntax

```
bgp log-neighbor-changes
no bgp log-neighbor-changes
default bgp log-neighbor-changes
```

Example

- This command configures the switch to generate a message when a BGP peer enters of exits the Established state.

```
switch(config)#router bgp 1
switch(config-router-bgp)#bgp log-neighbor-changes
switch(config-router-bgp)#
```

clear ip bgp

The **clear ip bgp** command removes BGP IPv4 learned routes from the routing table, reads all routes from designated peers, and sends routes to those peers as required.

- a hard reset tears down and rebuilds the peering sessions and rebuilds BGP routing tables.
- a soft reset uses stored prefix information to reconfigure and activate BGP routing tables without tearing down existing peering sessions.

  Soft resets use stored update information to apply new BGP policy without disrupting the network.

Routes that are read or sent are processed through modified route maps or AS-path access lists. The command can also clear the switch's BGP sessions with its peers.

After a route map is modified, the changes do not take effect until the BGP process is forced to recognize the changes. Use the **clear ip bgp** command after changing any of these BGP attributes:

- access lists
- weights
- distribution lists
- timers
- administrative distance
- route maps

Command Mode

Privileged EXEC

Command Syntax

```
clear ip bgp [ACTION] [RESET_TYPE] [DATA_FLOW]
```

Parameters

- *ACTION* the entity upon which the clearing action is taken. Options include:
  - <no parameter> clears the routing table, then reads in routes from designated peers.
  - ***** clears all BGP IPv4 sessions with the switch's peers.
  - *ipv4_addr* resets IPv4 session with peer at specified IPv4 address.
  - *ipv6_addr* resets IPv4 session with peer at specified IPv6 address
- *RESET_TYPE* reconfiguration type. Options include:
  - <no parameter> hard reset .
  - **soft** soft reset.
- *DATA_FLOW* restricts reset to inbound or outbound routes.
  - <no parameter> inbound and outbound routes are reset.
  - **in** inbound routes are reset.
  - **out** outbound routes are reset.

Examples

- This command removes all BGP learned routes from the routing table:

```
switch#clear ip bgp
switch#
```

- This command clears all of the switch's BGP IPv4 peering sessions:

```
switch#clear ip bgp *
switch#
```

clear ipv6 bgp

The **clear ipv6 bgp** command removes BGP IPv6 learned routes from the routing table, reads all routes from designated peers, and sends routes to those peers as required.

- a hard reset tears down and rebuilds the peering sessions and rebuilds BGP routing tables.
- a soft reset uses stored prefix information to reconfigure and activate BGP routing tables without tearing down existing peering sessions.

  Soft resets use stored update information to apply new BGP policy without disrupting the network.

Routes that are read or sent are processed through modified route maps or AS-path access lists. The command can also clear the switch's BGP sessions with its peers.

After a route map is modified, the changes do not take effect until the BGP process is forced to recognize the changes. Use the **clear ipv6 bgp** command after changing any of these BGP attributes:

- access lists
- weights
- distribution lists
- timers
- administrative distance
- route maps

Command Mode

Privileged EXEC

Command Syntax

```
clear ipv6 bgp [ACTION][RESET_TYPE][DATA_FLOW]
```

Parameters

- *ACTION* the entity upon which the clearing action is taken. Options include:
  - <no parameter> clears the routing table, then reads in routes from designated peers.
  - ***** clears all BGP IPv6 sessions with the switch's peers.
  - *ipv4_addr* resets IPv6 session with peer at specified IPv4 address.
  - *ipv6_addr* resets IPv6 session with peer at specified IPv6 address.
- *RESET_TYPE* reconfiguration type. Options include:
  - <no parameter> hard reset.
  - **soft** soft reset.
- *DATA_FLOW* restricts reset to inbound or outbound routes.
  - <no parameter> inbound and outbound routes are reset.
  - **in** inbound routes are reset.
  - **out** outbound routes are reset.

Examples

- This command removes all BGP IPv6 learned routes from the routing table:

```
switch#clear ipv6 bgp
switch#
```

- This command clears all of the switch's BGP IPv6 peering sessions:

```
switch#clear ip bgp *
switch#
```

distance bgp

The **distance bgp** command assigns an administrative distance to routes that the switch learns through BGP. Routers use administrative distances to select a route when two protocols provide routing information to the same destination. Distance values range from 1 to 255; lower distance values correspond to higher reliability. BGP routing tables do not include routes with a distance of 255.

The distance command assigns distance values to external, internal, and local BGP routes:

- **external**: External routes are routes for which the best path is learned from a neighbor external to the autonomous system. Default distance is 200.
- **internal**: Internal routes are routes learned from a BGP entity within the same autonomous system. Default distance is 200.
- **local**: Local routes are networks listed with a network router configuration command for that router or for networks that are redistributed from another process. Default distance is 200.

The **no distance bgp** and **default distance bgp** commands restore the default administrative distances by removing the **distance bgp** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
distance bgp external_dist [INTERNAL_LOCAL]
no distance bgp
default distance bgp
```

Parameters

- *external_dist* distance assigned to external routes. Values range from 1 to 255.
- ***INTERNAL_LOCAL*** distance assigned to internal and local routes. Values for both routes range from 1 to 255. Options include:
  - — <no parameter> *external_dist* value is assigned to internal and local routes.
  - — *internal_dist local_dist* values assigned to internal (*internal_dist)* and local (*local_dist*) routes.

Example

- This command assigns an administrative distance of 150 to external routes, 200 to internal, and 150 to local routes.

```
switch(config)#router bgp 1
switch(config-router-bgp)#distance bgp 150 200 150
switch(config-router-bgp)#
```

ip as-path access-list

The **ip as-path access-list** command creates an access list to filter BGP route updates. If access list ***list_name*** does not exist, this command creates it. If it already exists, this command appends statements to the list.

The **no ip as-path access-list** and **default ip as-path access-list** commands delete the named access list.

Command Mode

Global Configuration

Command Syntax

```
ip as-path access-list list_name FILTER_TYPE regex ORIGIN
no ip as-path access-list list_name
default ip as-path access-list list_name
```

Parameters

- *list_name* the name of the AS path access list.
- ***FILTER_TYPE*** access resolution of the specified community. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- *regex* a regular expression describing the AS path being filtered. Regular expressions are pattern matching strings that composed of text characters and operators (Section 3.2.6).
- ***ORIGIN*** the origin of the path information. Values include:
  - <no parameter> sets the origin to ***any***.
  - **any** any BGP origin.
  - **egp** EGP origin.
  - **igp** IGP origin.
  - **incomplete** incomplete origin.

Example

- These commands create an AS path access list named “list1” which allows all BGP routes except those originating in AS 3.

```
switch(config)#ip as-path access-list list1 deny _3$
switch(config)#ip as-path access-list list1 permit .*
switch(config)#
```

ip community-list expanded

The **ip community-list expanded** command creates and configures a BGP access list based on BGP communities. A BGP community access list filters route maps that are configured as BGP communities. The command uses regular expressions to name the communities specified by the list.

The **no ip community-list expanded** and **default ip community-list expanded** commands delete the specified community list by removing the corresponding **ip community-list expanded** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
ip community-list expanded listname FILTER_TYPE R_EXP
no ip community-list expanded listname
default community-list expanded listname
```

Parameters

- *listname* name of the community list. Valid input is text.
- ***FILTER_TYPE*** access resolution of the specified community. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- ***R_EXP*** list of communities, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators (Section 3.2.6)

Example

- This command creates a BGP community list that permits routes from networks 20-24 and 30-34 in autonomous system 10.

```
switch(config)#ip community-list expanded list_2 permit 10:[2-3][0-4]_
switch(config)#
```

ip community-list standard

The **ip community-list standard** command creates and configures a BGP access list based on BGP communities. A BGP community list filters route maps that are configured as BGP communities.

The **no ip community-list standard** and **default ip community-list standard** commands delete the specified community list by removing the corresponding **ip community-list standard** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
ip community-list standard listname FILTER_TYPE COMM_1 [COMM_2...COMM_n]
no ip community-list standard listname
default ip community-list standard listname
```

Parameters

- *listname* name of the community list. Valid input is text.
- *FILTER_TYPE* access resolution of the specified community. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- *COMM_x* community number or name, as specified in the route map that sets the community list number.
  - *aa:nn* AS and network number, separated by colon. Each value ranges from 1 to 65535.
  - *number* community number. Values range from 1 to 4294967040.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.

Example

- This command creates a BGP community list (named list_9) that denies members of route maps configured as AS-network number 100:250.

```
switch(config)#ip community-list standard list_9 deny 100:250
switch(config)#
```

ip extcommunity-list expanded

The **ip extcommunity-list expanded** command creates an extended community list to configure Virtual Private Network (VPN) route filtering. Extended community attributes filter routes for VPN routing and forwarding instances (VRFs). The command uses regular expressions to name the communities specified by the list.

- **Route Target (rt)** attribute identifies a set of sites and VRFs that may receive routes that are tagged with the configured route target. Configuring the route target extended attribute with a route allows that route to be placed in the per-site forwarding tables that route traffic received from corresponding sites.
- **Site of Origin (soo)** attribute uniquely identifies the site from which the provider edge (PE) router learned the route. All routes learned from a specific site must be assigned the same site of origin attribute whether a site is connected to a single PE router or multiple PE routers. Configuring this attribute prevents the creation of routing loops when a site is multihomed. The SOO extended community attribute is configured on the interface and is propagated into BGP through redistribution. The SOO should not be configured for stub sites or sites that are not multihomed.

The **no ip extcommunity-list expanded** and **default ip extcommunity-list expanded** commands delete the specified extended community list by removing the corresponding **ip community-list expanded** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip extcommunity-list expanded listname FILTER_TYPE R_EXP
no ip extcommunity-list expanded listname
default ip extcommunity-list expanded listname
```

Parameters

- *listname* name of the extended community list. Valid input is text.
- ***FILTER_TYPE*** access resolution of the specified extended community list. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- ***R_EXP*** list of communities, formatted as a regular expression. Regular expressions are pattern matching strings that are composed of text characters and operators.
  - Regular expressions that begin ***RT:*** match the ***rt*** ext. community attribute option
  - Regular expressions that begin ***SoO:*** match the ***soo*** ext. community attribute option.

  RT: and ***SoO:*** are case sensitive.

  Section 3.2.6 describes regular expressions.

Example

- This command creates a BGP extended community list that denies routes from route target networks 20-24 and 30-34 in autonomous system 10.

```
switch(config)#ip extcommunity-list expanded list_1 deny RT:10:[2-3][0-4]_
switch(config)#
```

ip extcommunity-list standard

The **ip extcommunity-list standard** command creates an extended community list to configure Virtual Private Network (VPN) route filtering. Extended community attributes filter routes for VPN routing and forwarding instances (VRFs).

- **Route Target (rt)** attribute identifies a set of sites and VRFs that may receive routes that are tagged with the configured route target. Configuring the route target extended attribute with a route allows that route to be placed in the per-site forwarding tables that route traffic received from corresponding sites.
- **Site of Origin (soo)** attribute uniquely identifies the site from which the provider edge (PE) router learned the route. All routes learned from a specific site must be assigned the same site of origin attribute whether a site is connected to a single PE router or multiple PE routers. Configuring this attribute prevents the creation of routing loops when a site is multihomed. The SOO extended community attribute is configured on the interface and is propagated into BGP through redistribution. The SOO should not be configured for stub sites or sites that are not multihomed.

The **no ip extcommunity-list standard** and **default ip extcommunity-list standard** commands delete the specified extended community list by removing the corresponding **ip extcommunity-list standard** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip extcommunity-list standard listname FILTER_TYPE COMM_1 [COMM_2...COMM_n]
no ip extcommunity-list standard listname
default ip extcommunity-list standard listname
```

Parameters

- *listname* name of the extended community list. Valid input is text.
- ***FILTER_TYPE*** access resolution of the specified extended community list. Options include:
  - **permit** access is permitted.
  - **deny** access is denied.
- ***COMM_x*** extended community attribute. Options include:
  - **rt** *aa:nn* route target, as specified by autonomous system:network number
  - **rt** *ip_addr:nn* route target, as specified by ip address:network number
  - **soo** *aa:nn* site of origin, as specified by autonomous system:network number
  - **soo** *ip_addr:nn* site of origin, as specified by ip address:network number

Example

- This command creates a BGP extended community list that denies routes from route target 100:250.

```
switch(config)#ip extcommunity-list standard list_9 deny rt 100:250
switch(config)#
```

maximum paths (BGP)

The **maximum-paths** command controls the maximum number of parallel eBGP routes that the switch supports. The default maximum is one route. The command provides an ECMP (equal cost multiple paths) parameter that controls the number of equal-cost paths that the switch stores in the routing table for each route.

The **no maximum-paths** and **default maximum-paths** commands restore the default values of the maximum number of parallel routes and the maximum number of ECMP paths by removing the corresponding command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
maximum-paths paths [ecmp ecmp_paths]
no maximum-paths
default maximum-paths
```

Parameters

- *paths* maximum number of parallel routes. Default value is 1.
- *ecmp_paths* maximum number of ECMP paths for each route. Default is maximum value.

  Values for each parameter range from 1 to the maximum number of interfaces per ECMP group. The maximum number of interfaces per ECMP group is platform dependent (Table 1-3).

Examples

- This command configures the maximum number of BGP parallel paths to 12. The ECMP value for each route is 16 (FM4000 or PetraA platforms) or 32 (Trident platform).

```
switch(config)#router bgp 1
switch(config-router-bgp)#maximum-paths 12
! Warning: maximum-paths will take effect after BGP restart.
switch(config-router-bgp)#
```

- This command configures the maximum number of BGP parallel paths to 2. The ECMP value for each route is 4.

```
switch(config)#router bgp 1
switch(config-router-bgp)#maximum-paths 2 ecmp 4
! Warning: maximum-paths will take effect after BGP restart.
switch(config-router-bgp)#
```

neighbor activate

The **neighbor activate** command defines the configuration mode address family activation state of a specified address that is configured as a BGP neighbor. The switch sends the following announcements to addresses active in an address family:

- IPv4 address family: IPv4 capability and all network advertisements with IPv4 prefixes.
- IPv6 address family: IPv6 capability and all network advertisements with IPv6 prefixes.

The **bgp default** command configures the default address family activation state of addresses configured as BGP neighbors. The **neighbor activate** and **no neighbor activate** commands override the neighbor's default activation state state within the configuration mode address family:

- **neighbor activate**: the specified address is active in the address family.
- **no neighbor activate**: the specified address is not active in the address family.

The **default neighbor activate** command removes the corresponding **neighbor activate** or **no neighbor activate** command from ***running-config***, restoring the default address family activation state for the specified neighbor address.

Command Mode

Router-BGP-Address-Family Configuration

Command Syntax

```
neighbor NEIGHBOR_ID activate
no neighbor NEIGHBOR_ID activate
default neighbor NEIGHBOR_ID activate
```

Parameters

- *NEIGHBOR_ID* IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.

Limitations

The switch supports the advertisement of networks with IPv6 prefixes to IPv4 transport neighbors. The switch does not support the advertisement of networks with IPv4 prefixes to IPv6 transport neighbors.

Example

- The two neighbor activation commands activate the advertising of specified neighbors during IPv4 peering sessions. The **show active** command displays the result of the previous commands.

```
switch(config)#router bgp 1
switch(config-router-bgp)#no address-family ipv4
switch(config-router-bgp-af)#neighbor 172.41.18.15 activate
switch(config-router-bgp-af)#neighbor 171.49.22.6 activate
switch(config-router-bgp-af)#no neighbor 171.15.21.18 activate
switch(config-router-bgp-af)#show active
   address-family ipv4
      no neighbor 171.15.21.18 activate
      neighbor 171.49.22.6 activate
      neighbor 172.41.18.15 activate
switch(config-router-bgp-af)#exit
switch(config-router-bgp)#
```

neighbor allowas-in

The **neighbor allowas-in** command configures the switch to permit the advertisement of prefixes containing duplicate autonomous switch numbers (ASNs). This command programs the switch to ignore its ASN in the AS path of routes and allow them into the routing domain. This function is disabled by default.

The **no neighbor allowas-in** and **default neighbor allowas-in** commands restore the default configuration by removing the corresponding **neighbor allowas-in** command from ***running-config***.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID allowas-in [asn_quantity]
no neighbor NEIGHBOR_ID allowas-in
default neighbor NEIGHBOR_ID allowas-in
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *asn_quantity* Number of switch's (ASN) allowed in path. Values range from 1 to 10. Default is 3.

Example

- This command activates the allowas-in function for the neighbor at 14.4.1.30.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 14.4.1.30 allowas-in
switch(config-router-bgp)#
```

neighbor description

The **neighbor description** command associates descriptive text with the specified peer.

The **no neighbor description** and **default neighbor description** commands remove the text association from the specified peer.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID description description_string
no neighbor NEIGHBOR_ID description
default neighbor NEIGHBOR_ID description
```

Parameters

- ***NEIGHBOR_ID*** IP address. Options include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
- *description_string* text string that is associated with neighbor.

Example

- This command associates the string PEER_1 with the peer located at 14.4.1.30.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 14.4.1.30 description PEER_1
switch(config-router-bgp)#
```

neighbor ebgp-multihop

The **neighbor ebgp-multihop** command programs the switch to accept and attempt BGP connections to the external peers residing on networks not directly connected to the switch. The command does not establish the multihop if the only route to the peer is the default route (0.0.0.0).

The **no neighbor ebgp-multihop** and **default neighbor ebgp-multihop** commands restore the default configuration by removing the corresponding **neighbor ebgp-multihop** command from ***running-config***.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID ebgp-multihop [hop_number]
no neighbor NEIGHBOR_ID ebgp-multihop
default neighbor NEIGHBOR_ID ebgp-multihop
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *hop_number* time-to-live (hops). Values range from 1 to 255. Default value is 255.

Example

- This command programs the switch to accept and attempt BGP connections to the external peer located at 14.4.1.30, setting the hop limit to 32.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 14.4.1.30 ebgp-multihop 32
switch(config-router-bgp)#
```

neighbor export-localpref

The **neighbor export-localpref** command determines the LOCAL_PREF value that is sent in BGP UPDATE packets to the specified peer. This command has no effect on external peers.

The **no neighbor export-localpref** and **default neighbor export-localpref** commands reset the LOCAL_PREF value to the default of 100 in packets sent to the specified peer.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID export-localpref preference
no neighbor NEIGHBOR_ID export-localpref
default neighbor NEIGHBOR_ID export-localpref
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
- *preference* preference value. Values range from 0 to 4294967295.

Example

- This command configures the switch to fill the LOCAL_PREF field with 200 in UPDATE packets that it sends to the peer located at 10.1.1.45.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 10.1.1.45 export-localpref 200
switch(config-router-bgp)#
```

neighbor import-localpref

The **neighbor import-localpref** command determines the local preference assigned to routes received from the specified external peer. This command has no effect on routes received from internal peers.

The **no neighbor import-localpref** and **default neighbor import-localpref** commands reset the local preference to the default of 100 for routes received from the specified peer.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID import-localpref preference
no neighbor NEIGHBOR_ID import-localpref
default neighbor NEIGHBOR_ID import-localpref
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *preference* preference value. Values range from 0 to 4294967295.

Example

- This command configures the switch to assign a local preference of 50 to routes received from the peer located at 14.4.1.30.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 14.4.1.30 import-localpref 50
switch(config-router-bgp)#
```

neighbor local-as

The **neighbor local-as** command enables the modification of the AS_PATH attribute for routes received from an eBGP neighbor, allowing the switch to appear as a member of a different autonomous system (AS) to external peers. This switch does not prepend the local AS number to routes received from the eBGP neighbor. The AS number from the local BGP routing process is not prepended.

The **no neighbor local-as** and **default neighbor local-as** commands disable AS_PATH modification by removing the **neighbor local-as** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID local-as as_id no-prepend replace-as
no neighbor NEIGHBOR_ID local-as
default neighbor NEIGHBOR_ID local-as
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *as_id* AS number that is prepended to the AS_PATH attribute. Values range from 1 to 65535.

  This parameter cannot be set to AS numbers from the local BGP routing process or the network of the remote peer.

Example

- For the neighbor at 10.13.64.1, these commands remove AS 300 from outbound routing updates and replace it with AS 600.

```
switch(config)#router bgp 300
switch(config-router-bgp)#neighbor 10.13.64.1 600
switch(config-router-bgp)#
```

neighbor local-v6-addr

The **neighbor local-v6-addr** command specifies the next-hop value that the switch sends as the IPv6 NLRI value to neighbors with whom IPv4 transport peering is established.

In IPv6 peering sessions, the switch sends the global IPv6 address of the interface that is used to transmit BGP updates.

The **no neighbor local-v6-addr** and **default neighbor local-v6-addr** commands delete the corresponding **neighbor local-v6-addr** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID local-v6-addr ipv6_local
no neighbor NEIGHBOR_ID local-v6-addr
default neighbor NEIGHBOR_ID local-v6-addr
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *group_name* peer group name.
- *ipv6_local* Next hop address (A:B:C:D:E:F:G:H).

Example

- For the neighbor at 172.15.21.18, these commands specify an IPv6 NLRI value that is sent during IPv4 transport peering sessions.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 171.15.21.18 local-v6-addr fd7a:71bf:c2a4:::2
switch(config-router-bgp)#show active
router bgp 1
   bgp log-neighbor-changes
   bgp default ipv6-unicast
   neighbor 171.15.21.18 local-v6-addr fd7a:71bf:c2a4:1761::2
switch(config-router-bgp)#
```

neighbor maximum-routes

The **neighbor maximum-routes** command determines the number of BGP routes the switch accepts from a specified neighbor and defines an action when the limit is exceeded. The default value is 12,000. To remove the maximum routes limit, specify a limit of zero.

If the number of routes received from a peer exceeds this, the switch generates an error message. This command can also configure the switch to disable peering with the neighbor – in this case, the neighbor state is reset only through a **clear ip bgp** command.

The **no neighbor maximum-routes** and **default neighbor maximum-routes** commands reset the maximum-routes value to the default value of 12,000 for the specified peer.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID maximum-routes quantity [ACTION]
no neighbor NEIGHBOR_ID maximum-routes
default neighbor NEIGHBOR_ID maximum-routes
```

Parameters

- *NEIGHBOR_ID* IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *quantity* maximum number of routes. Values include:
  - 0 the switch does not define a route limit.
  - 1 to 4294967294 maximum number of routes.
- *ACTION* switch action when the route limit is exceeded. Values include:
  - <no parameter> peering is disabled and an error message is generated.
  - **warning-only** peering is not disabled, but an error message is generated.

Example

- This command configures the switch to accept 15000 routes for the neighbor at 12.12.18.240. If the neighbor exceeds 15000 routes, the switch disables peering with the neighbor.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 12.12.18.240 maximum-routes 15000
switch(config-router-bgp)#
```

neighbor next-hop-peer

The **neighbor next-hop-peer** command configures the switch to list the peer address as the next hop in routes that it receives from the specified peer BGP-speaking neighbor. This command overrides the next hop for all routes received from this neighbor.

The **no neighbor next-hop-peer** and **default neighbor next-hop-peer** commands remove the next hop configuration for the specified neighbor by removing the corresponding **neighbor next-hop-peer** command from ***running-config***.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID next-hop-peer
no neighbor NEIGHBOR_ID next-hop-peer
default neighbor NEIGHBOR_ID next-hop-peer
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.

Example

- This command configures the peer address of 14.15.2.24 as the next hop for routes advertised to the switch from the peer BGP speaking neighbor.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 14.15.2.24 next-hop-peer
switch(config-router-bgp)#
```

neighbor next-hop-self

The **neighbor next-hop-self** command configures the switch to list its address as the next hop in routes that it advertises to the specified BGP-speaking neighbor. This function is used in networks where BGP neighbors do not directly access all other neighbors on the same subnet.

The **no neighbor next-hop-self** and **default neighbor next-hop-self** commands remove the next hop configuration for the specified neighbor by removing the corresponding **neighbor next-hop-self** command from ***running-config***.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID next-hop-self
no neighbor NEIGHBOR_ID next-hop-self
default neighbor NEIGHBOR_ID next-hop-self
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.

Example

- This command configures the switch as the next hop for the peer at 14.4.1.30.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 14.4.1.30 next-hop-self
switch(config-router-bgp)#
```

neighbor out-delay

The **neighbor out-delay** command sets the period that a route update for a specified neighbor must be in the routing table before the switch exports it to BGP. The out delay interval is used for bundling routing updates.

The **no neighbor out-delay** and **default neighbor out-delay** commands restore the default out delay value of zero by deleting the corresponding **neighbor out-delay** command from ***running-config*.**

The **no neighbor** command removes all configuration commands for the specified neighbor.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID out-delay delay_time
no neighbor NEIGHBOR_ID out-delay
default neighbor NEIGHBOR_ID out-delay
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *delay_time* the out delay period (seconds) . Values range from 0 to 600. Default value is 0.

Example

- This command sets the out delay period to 5 seconds for the connection with the peer at 10.24.15.9.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 10.24.15.9 out-delay 5
switch(config-router-bgp)#
```

neighbor password

The **neighbor password** command enables authentication on a TCP connection with a BGP peer. The plain-text version of the password is a string, up to 8 bytes in length. Peers must use the same password to ensure proper communication.

Running-config displays the encrypted version of the password. The encryption scheme is not strong by cryptographic standards; encrypted passwords should be treated in the same manner as plain-text passwords.

The **no neighbor password** and **default neighbor password** commands remove the neighbor password from the configuration, disabling authentication with the specified peer.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID password [ENCRYPT_LEVEL] key_text
no neighbor NEIGHBOR_ID password
default neighbor NEIGHBOR_ID password
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
- ***ENCRYPT_LEVEL*** the encryption level of the *key_text* parameter. Values include:
  - <no parameter> indicates the *key_text* is in clear text.
  - **0** indicates *key_text* is in clear text. Equivalent to the <no parameter> case.
  - **7** indicates *key_text* is md5 encrypted.
- *key_text* the password.

Example

- This command specifies a password in clear text.

```
switch(config)#router bgp 1
switch(config-router-bgp)#neighbor 10.25.25.13 password 0 code123
switch(config-router-bgp)#
```

Running-config stores the password as an encrypted string.

neighbor peer-group (create)

Peer groups allow the user to apply settings to a group of BGP neighbors simultaneously. Once a peer group is created, the group name can be used as a parameter in neighbor configuration commands, and the configuration will be applied to all members of the group. Settings applied to an individual neighbor in the peer group override group settings.

The **neighbor peer-group (create)** command is used to create BGP peer groups. To assign BGP neighbors to those peer groups, use the **neighbor peer-group (neighbor assignment)** command.

The **no neighbor peer-group (create)** and **default neighbor peer-group (create)** commands remove the specified peer group from ***running-config***. When a peer group is deleted, the neighbors that were members of that peer group retain the configuration inherited from the peer group.

The **no neighbor** command removes all configuration commands for the specified neighbor.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor group_name peer-group
no neighbor group_name peer-group
default neighbor group_name peer-group
```

Parameters

- *group_name* peer group name.

Examples

- These commands create a BGP peer group called bgpgroup1, assign several neighbors to the group, and apply a route map.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor bgpgroup1 peer-group
switch(config-router-bgp)#neighbor 1.1.1.1 peer-group bgpgroup1
switch(config-router-bgp)#neighbor 2.2.2.2 peer-group bgpgroup1
switch(config-router-bgp)#neighbor 3.3.3.3 peer-group bgpgroup1
switch(config-router-bgp)#neighbor bgpgroup1 route-map corporate in
switch(config-router-bgp)#show active
router bgp 9
bgp log-neighbor-changes
   neighbor bgpgroup1 peer-group
   neighbor bgpgroup1 route-map corporate in
   neighbor bgpgroup1 maximum-routes 12000
   neighbor 1.1.1.1 peer-group bgpgroup1
   neighbor 1.1.1.1 maximum-routes 12000
   neighbor 2.2.2.2 peer-group bgpgroup1
   neighbor 2.2.2.2 maximum-routes 12000
   neighbor 3.3.3.3 peer-group bgpgroup1
   neighbor 3.3.3.3 maximum-routes 12000
switch(config-router-bgp)#
```

- This command removes peer group "bgpgroup1" from ***running-config***. All settings that group members inherited from the peer group are maintained.

```
switch(config-router-bgp)#no neighbor bgpgroup1 peer-group
switch(config-router-bgp)#show active
router bgp 9
   bgp log-neighbor-changes
   neighbor 1.1.1.1 route-map corporate in
   neighbor 1.1.1.1 maximum-routes 12000
   neighbor 2.2.2.2 route-map corporate in
   neighbor 2.2.2.2 maximum-routes 12000
   neighbor 3.3.3.3 route-map corporate in
   neighbor 3.3.3.3 maximum-routes 12000
switch(config-router-bgp)#
```

neighbor peer-group (neighbor assignment)

Peer groups allow the user to apply settings to a group of BGP neighbors simultaneously. Once a peer group is created, the group name can be used as a parameter in neighbor configuration commands, and the configuration will be applied to all members of the group. Settings applied to an individual neighbor in the peer group override group settings.

The **neighbor peer-group (neighbor assignment)** command is used to assign BGP neighbors to an existing peer group. To create a peer group, use the **neighbor peer-group (create)** command.

The **no neighbor peer-group (neighbor assignment)** and **default neighbor peer-group (neighbor assignment)** commands remove the specified neighbor from all peer groups. When a neighbor is removed from a peer group, the neighbor retains the configuration inherited from the peer group.

The **no neighbor** command removes all configuration commands for the specified neighbor.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor ipv4_address peer-group group_name
no neighbor ipv4_address peer-group
default neighbor ipv4_address peer-group
```

Parameters

- *ipv4_address* neighbor's IPv4 address.
- *group_name* peer group name.

Examples

- These commands create a BGP peer group called bgpgroup1, assign several neighbors to the group, and apply a route map.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor bgpgroup1 peer-group
switch(config-router-bgp)#neighbor 1.1.1.1 peer-group bgpgroup1
switch(config-router-bgp)#neighbor 2.2.2.2 peer-group bgpgroup1
switch(config-router-bgp)#neighbor 3.3.3.3 peer-group bgpgroup1
switch(config-router-bgp)#neighbor bgpgroup1 route-map corporate in
switch(config-router-bgp)#
```

- This command removes the neighbor at 1.1.1.1 from the peer group. All settings that neighbor 1.1.1.1 inherited from the peer group are maintained.

```
switch(config-router-bgp)#no neighbor 1.1.1.1 peer-group
switch(config-router-bgp)#
```

neighbor remote-as

The **neighbor remote-as** command establishes a neighbor (peer) connection. Internal neighbors have the same AS number. External neighbors have different AS numbers.

The **no neighbor remote-as** and **default neighbor remote-as** commands disable peering with the specified address by removing the corresponding **neighbor remote-as** command from ***running-config***.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID remote-as as_id
no neighbor NEIGHBOR_ID remote-as
default neighbor NEIGHBOR_ID remote-as
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *as_id* Autonomous system (AS) of the peer. Values range from 1 to 65535.

Example

- This command establishes a BGP connection with the router at 16.2.29.14 in AS 300.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 16.2.29.14 remote-as 300
switch(config-router-bgp)#
```

neighbor remove-private-as

The **neighbor remove-private-as** command removes private autonomous system numbers from outbound routing updates for external BGP (eBGP) neighbors. When the autonomous system path includes both private and public autonomous system numbers, the private autonomous system number is not removed.

The **no neighbor remove-private-as** and **default neighbor remove-private-as** commands restore the default behavior by removing the **neighbor remove-private-as** statement from ***running-config***.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID remove-private-as
no neighbor NEIGHBOR_ID remove-private-as
default neighbor NEIGHBOR_ID remove-private-as
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.

Example

- This command programs the switch to remove private AS numbers from outbound routing updates for the eBGP neighbor at 16.2.29.14.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 34.2.29.14 remove-private-as
switch(config-router-bgp)#
```

neighbor route-map (BGP)

The **neighbor route-map** command applies a route map to inbound or outbound BGP routes. When a route map is applied to outbound routes, the switch will advertise only routes matching at least one section of the route map. Only one outbound route map and one inbound route map can be applied to a given neighbor. A new route map applied to a neighbor will replace the previous route map.

The command is available in Router-BGP and Router-BGP-Address-Family configuration modes. The mode in which the command is executed determines the scope of the command:

- In Router-BGP mode, the route map is applied to specified neighbor in all peering sessions where it is advertised.
- In Router-BGP-Address-Family mode, the route map is applied to the neighbors only in peering sessions corresponding to the configuration mode address family.

The **no neighbor route-map** and **default neighbor route-map** commands discontinue the application of the specified route map to the specified routes by deleting the corresponding **neighbor route-map** command from ***running-config***. Removing a route map from one direction does not remove it from the other if it has been applied to both.

Command Mode

Router-BGP Configuration
Router-BGP Configuration-Address-Family

Command Syntax

```
neighbor NEIGHBOR_ID route-map map_name DIRECTION
no neighbor NEIGHBOR_ID route-map map_name DIRECTION
default neighbor NEIGHBOR_ID route-map map_name DIRECTION
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *map_name* name of a route map.
- ***DIRECTION*** routes to which the route map is applied. Options include:
  - **in** route map is applied to inbound routes.
  - **out** route map is applied to outbound routes.

Example

- This command applies a route map named ***inner-map*** to a BGP inbound route from 101.72.14.5.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 101.72.14.5 route-map inner-map in
switch(config-router-bgp)#
```

neighbor route-reflector-client

Participating BGP routers within an AS communicate EBGP-learned routes to all of their peers, but to prevent routing loops they must not re-advertise IBGP-learned routes within the AS. To ensure that all members of the AS share the same routing information, a fully meshed network topology (in which each member router of the AS is connected to every other member) can be used, but this topology can result in high volumes of IBGP messages when it is scaled. Instead, in larger networks one or more routers can be configured as route reflectors.

A route reflector is configured to re-advertise routes learned through IBGP to a group of BGP neighbors within the AS (its clients), eliminating the need for a fully meshed topology.

The **neighbor route-reflector-client** command configures the switch to act as a route reflector and configures the specified neighbor as one of its clients. Additional clients can be specified by re-issuing the command.

The **bgp client-to-client reflection** command controls client-to-client reflection.

The **no neighbor route-reflector-client** and **default neighbor route-reflector-client** commands disable route refection by deleting the **neighbor route-reflector-client** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID route-reflector-client
no neighbor NEIGHBOR_ID route-reflector-client
default neighbor NEIGHBOR_ID route-reflector-client
```

Parameters

- ***NEIGHBOR_ID*** IP address of neighbor. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *group_name* peer group name.

Example

- This command configures the switch as a route reflector and the neighbor at 101.72.14.5 as one of its clients.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 101.72.14.5 route-reflector-client
switch(config-router-bgp)#
```

neighbor send-community

The **neighbor send-community** command configures the switch to send community attributes to the specified BGP neighbor.

The **no neighbor send-community** and **default neighbor send-community** commands discontinue the sending of community attributes to the specified neighbor by deleting the corresponding **neighbor send-community** statement from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID send-community
no neighbor NEIGHBOR_ID send-community
default neighbor NEIGHBOR_ID send-community
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.

Example

- This command configures the switch to send community attributes to the neighbor at address 10.5.2.23.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.5.2.23 send-community
switch(config-router-bgp)#
```

neighbor shutdown

The **neighbor shutdown** command disables the specified neighbor. Disabling a neighbor also terminates all of its active sessions and removes associated routing information.

The **no neighbor shutdown** and **default neighbor shutdown** commands enable the specified neighbor by removing the corresponding **neighbor shutdown** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID shutdown
no neighbor NEIGHBOR_ID shutdown
default neighbor NEIGHBOR_ID shutdown
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.

Example

- This command applies a route map named ***inner-map*** to a BGP inbound route from 101.72.14.5.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 101.72.14.5 route-map inner-map in
switch(config-router-bgp)#
```

neighbor soft-reconfiguration

By default, inbound BGP routes which are filtered out by the switch's import policy are still stored on the switch. Because all routes are retained, this allows policies to be changed without resetting BGP sessions. It also allows the switch to display all advertised routes when the **show ip bgp neighbor advertised-routes** command is issued.

The **no neighbor soft-reconfiguration** command configures the switch to discard information about routes that fail the import policy.

The **neighbor soft-reconfiguration** and **default neighbor soft-reconfiguration** commands restore the default behavior by removing the **no neighbor soft-reconfiguration** statement from ***running-config***. Only the **no** form of this command is visible in ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID soft-configuration inbound
no neighbor NEIGHBOR_ID soft-configuration inbound
default neighbor NEIGHBOR_ID soft-configuration inbound
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.

Example

- This command configures the switch to discard information about routes from the neighbor at 10.5.2.23 which are filtered out by the switch's import policies.

```
switch(config)#router bgp 9
switch(config-router-bgp)#no neighbor 10.5.2.23 soft-reconfiguration inbound
switch(config-router-bgp)#
```

neighbor timers

The **neighbor timers** command configures the BGP keepalive and hold times for a specified peer connection. The **timers bgp** command configures the times on all peer connection for which an individual command is not specified.

- Keepalive time is the period between the transmission of consecutive keepalive messages.
- Hold time is the period the switch waits for a KEEPALIVE or UPDATE message before it disables peering.

The hold time must be at least 3 seconds and should be three times longer than the keepalive setting.

The **no neighbor timers** and **default neighbor timers** commands remove the **neighbor timers** command from the configuration. The peer connection uses the timers specified by the **timers bgp** command.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID timers keep_alive hold_time
no neighbor NEIGHBOR_ID timers
default neighbor NEIGHBOR_ID timers
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *keep_alive* keepalive period, in seconds. Values include
  - **0** keepalive messages are not sent
  - **1** to **3600** keepalive time (seconds).
- *hold_time* hold time. Values include
  - **0** peering is not disabled by timeout expiry; keepalive packets are not sent.
  - **3** to **7200** hold time (seconds).

Example

- This command sets the keepalive time to 30 seconds and the hold time to 90 seconds for the connection with the peer at 10.24.15.9.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.24.15.9 timers 30 90
switch(config-router-bgp)#
```

neighbor update-source

The **neighbor update-source** command specifies the interface that BGP sessions use for TCP connections. By default, BGP sessions use the neighbor's closest interface (also known as the best local address).

The **no neighbor update-source** and **default neighbor update-source** commands restore the default setting by removing the **neighbor update-source** command from ***running-config***.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID update-source INTERFACE
no neighbor NEIGHBOR_ID update-source
default neighbor NEIGHBOR_ID update-source
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *group_name* peer group name.
- ***INTERFACE*** Interface type and number. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* loopback interface specified by *l_num*.
  - **management** *m_num* management interface specified by *m_num*.
  - **port-channel** *p_num* port channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command configures the switch to use Ethernet interface 10 for TCP connections for the neighbor at 14.4.1.30.

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 14.4.1.30 update-source ethernet 10
switch(config-router-bgp)#
```

neighbor weight

The **neighbor weight** command assigns a weight attibute value to paths from the specified neighbor. Weight is the first parameter that the BGP best-path selection algorithm considers. When multiple paths to a destination prefix exist, the best-path selection algorithm prefers the path with the highest weight. Other attributes are used only when all paths to the prefix have the same weight.

Weight values range from 0 to 65535 and are not propagated to other switches through route updates. The default weight for paths that the router originates is 32768; the default weight for routes received through BGP is 0.

A path's BGP weight is also configurable through route maps. Weight values set through route map commands have precedence over neighbor weight command values.

The **no neighbor weight** and **default neighbor weight** commands restore the default setting by removing the corresponding **neighbor weight** command from ***running-config***.

The **no neighbor** command removes all configuration commands for the neighbor at the specified address.

Command Mode

Router-BGP Configuration

Command Syntax

```
neighbor NEIGHBOR_ID weight weight_value
no neighbor NEIGHBOR_ID weight
default neighbor NEIGHBOR_ID weight
```

Parameters

- ***NEIGHBOR_ID*** IP address or peer group name. Values include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.
  - *group_name* peer group name.
- *weight_value* weight value. Values range from 1 to 65535.

Example

- This command specifies a weight of 4000 for all paths from the neighbor at 10.1.2.5

```
switch(config)#router bgp 9
switch(config-router-bgp)#neighbor 10.1.2.5 weight 4000
switch(config-router-bgp)#
```

network (BGP)

The **network** command specifies a network for advertisement through UPDATE packets to BGP peers. The configuration zeros the host portion of the specified network address; for example, 192.0.2.4/24 is stored as 192.0.2.0/24. A route map option is available for assigning attributes to the network.

The command is available in Router-BGP and Router-BGP-Address-Family configuration modes. The mode in which the command is issued does not affect the command. The scope of the command depends on the specified network address:

- Commands with an IPv4 address are advertised to peers activated in the IPv4 address family.
- Commands with an IPv6 address are advertised to peers activated in the IPv6 address family.

The **no network** and **default network** commands remove the network from the routing table, preventing its advertisement.

Command Mode

Router-BGP Configuration
Router-BGP Configuration-Address-Family

Command Syntax

```
network NET_ADDRESS [ROUTE_MAP]
no network NET_ADDRESS
default network NET_ADDRESS
```

Parameters

- ***NET_ADDRESS*** IP address range. Entry options include:
  - *ipv4_subnet* IPv4 subnet (CIDR notation).
  - *ipv4_addr* **mask** *subnet* IPv4 subnet (address-mask notation).
  - *ipv6_prefix* neighbor's IPv6 prefix (CIDR notation).
- ***ROUTE_MAP*** specifies route map that assigns attribute values to the network. Options include:
  - <no parameter> attributes are not assigned through a route map.
  - **route-map** *map_name* attributes listed by specified route map are assigned to the network.

Example

- This command enables BGP advertising for the network located at 14.5.8.23/24. The configuration stores the network as 14.5.8.0/24.

```
switch(config)#router bgp 9
switch(config-router-bgp)#network 14.5.8.23/24
switch(config-router-bgp)#
```

no neighbor

The **no neighbor** command removes all neighbor configuration commands for the specified neighbor. Commands removed by the **no neighbor** command include:

- **neighbor description**
- **neighbor ebgp-multihop**
- **neighbor export-localpref**
- **neighbor import-localpref**
- **neighbor local-as**
- **neighbor maximum-routes**
- **neighbor next-hop-peer**
- **neighbor next-hop-self**
- **neighbor out-delay**
- **neighbor password**
- **neighbor peer-group (create)**
- **neighbor peer-group (neighbor assignment)**
- **neighbor remote-as**
- **neighbor remove-private-as**
- **neighbor route-map (BGP)**
- **neighbor route-reflector-client**
- **neighbor send-community**
- **neighbor timers**
- **neighbor update-source**

Neighbor settings can be removed individually; refer to the command description page of the desired command for details. Neighbor settings for a peer group must be removed individually.

Command Mode

Router-BGP Configuration

Command Syntax

```
no neighbor NEIGHBOR_ID
default neighbor NEIGHBOR_ID
```

Parameters

- *NEIGHBOR_ID* IP address. Options include:
  - *ipv4_addr* neighbor's IPv4 address.
  - *ipv6_addr* neighbor's IPv6 address.

Example

- This command removes all neighbor configuration commands for the neighbor at 42.1.1.1.

```
switch(config)#router bgp 9
switch(config-router-bgp)#no neighbor 42.1.1.1
switch(config-router-bgp)#
```

redistribute (BGP)

The **redistribute** command enables the redistribution of specified routes to the BGP domain.

The **no redistribute** and **default redistribute** commands disable route redistribution from the specified domain by removing the corresponding **redistribute** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
redistribute ROUTE_TYPE [ROUTE_MAP]
no redistribute ROUTE_TYPE
default redistribute ROUTE_TYPE
```

Parameters

- ***ROUTE_TYPE*** source from which routes are redistributed. Options include:
  - **aggregate** BGP aggregate routes.
  - **connected** routes that are established when IP is enabled on an interface.
  - **OSPF** routes from an OSPF domain.
  - **OSPF match external** Routes external to the AS, but imported from OSPF.
  - **OSPF match internal** OSPF routes that are internal to the AS.
  - **RIP** routes from a RIP domain.
  - **static** IP static routes.
- ***ROUTE_MAP*** route map that determines the routes that are redistributed. Options include:
  - <no parameter> all routes are redistributed.
  - **route-map** *map_name* only routes in the specified route map are redistributed.

Example

- This command redistributes OSPF routes into the BGP domain.

```
switch(config)#router bgp 9
switch(config-router-bgp)#redistribute OSPF
switch(config-router-bgp)#
```

router-id (BGP)

The **router-id** command configures a fixed router ID for the local Border Gateway Protocol (BGP) routing process.

When the **router-id** command is not configured, the local router ID is set to the following:

- The loopback IP address when a loopback interface is configured.

  The loopback with the highest IP address is selected when multiple loopback interfaces are configured.

- The highest IP address on a physical interface when no loopback interfaces are configured.

Important The router-id must be specified if the switch has no IPv4 addresses configured.

The **no router-id** and **default router-id** commands remove the **router-id** command from ***running-config***.

Command Mode

Router-BGP Configuration

Command Syntax

```
router-id id_num
no router-id [id_num]
default router-id [id_num]
```

Parameters

- *id_num* router ID number (32-bit dotted decimal notation).

Example

- This command configures the fixed router ID address of 172.68.4.11

```
switch(config)#router bgp 9
switch(config-router-bgp)#router-id 172.68.4.11
switch(config-router-bgp)#
```

router bgp

The **router bgp** command places the switch in router-BGP configuration mode. If BGP was not previously instantiated, this command creates a BGP instance with the specified AS number. Router-BGP configuration mode is not a group change mode; ***running-config*** is changed immediately after commands are executed. The **exit** command does not affect the configuration.

When a BGP instance exists, the command must include the AS number of the existing BGP instance. Running this command with a different AS number generates an error message.

The **no router bgp** and **default router bgp** commands delete the BGP instance.

Refer to Router-BGP Configuration Mode (Includes Address-Family Mode) (page 904) for a list of commands available in router-BGP configuration mode.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
router bgp as_id
no router bgp
default router bgp
```

Parameters

- *as_id* Autonomous system (AS) number. Values range from 1 to 65535.

Examples

- This command creates a BGP instance with AS number 200.

```
switch(config)#router bgp 200
switch(config-router-bgp)#
```

- This command attempts to open a BGP instance with a different AS number from that of the existing instance. The switch displays an error and stays in global configuration mode.

```
switch(config)#router bgp 100
% BGP is already running with AS number 200
switch(config)#
```

- This command exits BGP configuration mode.

```
switch(config-router-bgp)#exit
switch(config)#
```

- This command deletes the BGP instance.

```
switch(config)#no router bgp
switch(config)#
```

show ip as-path access-list

The **show ip as-path access-list** command displays BGP filters on the switch. Specifying an access list displays the statements from that access list. Entering the command without parameters displays the statements from all access lists on the switch.

Command Mode

EXEC

Command Syntax

show ip as-path access-list [*list_name*]

Parameters

- *list_name* the name of an AS path access list.

Example

- This command displays the contents of the AS path access list named "list1."

```
switch>show ip as-path access-list list1
ip as-path access-list list1 deny _3$
ip as-path access-list list1 permit .*
switch>
```

show ip bgp

The **show ip bgp** command displays Border Gateway Protocol (BGP) IPv4 routing table entries. The output format depends on the command parameters:

- Data block format displays comprehensive information for each specified BGP routing table entry.
- Tabular format displays routing table entries in tabular format for the specified IPv4 addresses.

Command Mode

EXEC

Command Syntax

show ip bgp [***FILTER***]

Parameters

- ***FILTER*** routing table entries that the command displays. Values include*:*
  - — <no parameter> displays all routing table entries. Tabular format.
  - — **detail** displays all routing table entries. Data block format.
  - — *ipv4_addr* IPv4 host address. Data block format.
  - — *ipv4_subnet* IPv4 subnet address. (CIDR notation). Data block format.
  - — *ipv4_subnet* **longer-prefixes** IPv4 subnet address. (CIDR notation). Tabular format.

Examples

- This command displays the BGP routing table in the 172.17.48.0/23 network.

```
switch>show ip bgp 172.17.48.0/23
BGP routing table entry for 172.17.48.0/23
 Paths: 2 available
  (65533) 65534
    172.17.254.78 from 172.17.254.78 (172.26.0.34)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: 0:10
  (65533) 65534
    172.17.254.82 from 172.17.254.2 (172.26.0.23)
      Origin IGP, metric 0, localpref 100, valid, internal
      Router-ID: 172.26.0.23
switch>
```

- This command displays the BGP routing table.

```
switch>show ip bgp
Route status codes: s - suppressed, * - valid, > - active, e - ECMP

     Network             Next Hop        R Metric  LocPref Path
* >  0.0.0.0/0           -               u 10      4       i (Id 1)
* >  172.17.48.0/23      172.17.254.78   u 0       100     (65533) 65534 i (Id 8)
*    172.17.48.0/23      172.17.254.82   u 0       100     (65533) 65534 i (Id 7)
Rt-ID: 172.26.0.23
* >  172.17.50.0/23      172.17.254.78   u 0       100     (65533) 65534 i (Id 9)
*    172.17.50.0/23      172.17.254.82   u 0       100     (65533) 65534 i (Id 7)
Rt-ID: 172.26.0.23
switch>
```

show ip bgp community

The **show ip bgp community** command displays Border Gateway Protocol (BGP) routing table entries, filtered by community.

Command Mode

EXEC

Command Syntax

show ip bgp community [***COMM_1 ... COMM_n***] [***MATCH_TYPE***] [***DATA_OPTION***]

Parameters

- ***COMM_x*** community number or name, as specified in the route map that sets the community list number.
  - *aa:nn* AS and network number, separated by colon. Each value ranges from 1 to 65535.
  - *comm_num* community number. Values range from 1 to 4294967040.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- ***MATCH_TYPE*** Routes are filtered based on their communities.
  - <no parameter> routes must match at least one community in the list
  - **exact** route must match all communities and include no other communities.
- ***DATA_OPTION*** Type of information the command displays. Values include:
  - <no parameter> Displays table of of routing entry line items.
  - **detail** Displays data block for each routing table entry.

Example

- This command displays the BGP routing table entries for a specified community.

```
switch>show ip bgp community 65533:100 exact detail
BGP routing table entry for 172.17.254.0/30
 Paths: 1 available
  Local
    - from - (172.26.0.23)
      Origin IGP, metric 1, localpref 0, valid, local, best
      Community: 65533:100
switch>
```

show ip bgp neighbors

The **show ip bgp neighbors** command displays Border Gateway Protocol (BGP) and TCP session data for a specified IPv4 BGP neighbor. Command displays data for all neighbors if an address is not included.

Command Mode

EXEC

Command Syntax

show ip bgp neighbors [***NEIGHBOR_ADDR***]

Parameters

- *NEIGHBOR_ADDR* location of neighbors. Options include:
  - <no parameter> command displays information for all neighbors.
  - *ipv4_addr* command displays information for specified neighbor.

Related Command

show ip bgp neighbors (route type)
show ip bgp neighbors (route-type) community

Example

- This command displays information for the neighbor at 10.100.100.2

```
switch>show ip bgp neighbors 10.100.100.2
BGP neighbor is 10.100.100.2, remote AS 100
BGP version is 4, remote router ID 192.168.104.2
Negotiated version is 4
TTL is 0
holdtime is 90
restart-time is 0
Restarting: no
Current state is Established
Updates received: 1
Updates sent: 4
Total messages received: 372
Total messages sent: 383
Last state was OpenConfirm
Last event was RecvKeepAlive
Last error code was 0
Last error subcode was 0
Local TCP address is 10.100.100.1
Local AS is 100
Local router ID is 192.168.103.1
Capabilities                      Snt   Rcv   Neg
------------------------------------------------
Multiprotocol IPv4 Unicast        yes   yes   yes
Graceful Restart IPv4 Unicast     no    no    no
Multiprotocol IPv4 Multicast      no    no    no
Graceful Restart IPv4 Multicast   no    no    no
Route Refresh                     no    no    no
Send End-of-RIB messages          no    no    no
Dynamic Capabilities              no    no    no
switch>
```

show ip bgp neighbors (route type)

The **show ip bgp neighbors (route type)** command displays information for next hop routes to a specified IPv4 neighbor. The **show ip bgp neighbors (route-type) community** command displays the same information for routes filtered by communities.

The output format depends on the selected ***FILTER*** parameter:

- Data block format displays comprehensive information for each specified route.
- Tabular format displays routing table entries in tabular format for the specified IP addresses.

Commands that do not include a route type revert to the **show ip bgp neighbors** command.

Command Mode

EXEC

Command Syntax

show ip bgp neighbors *neighbor_addr* ***RTE*** [***FILTER***]

Related Command

show ip bgp neighbors
show ip bgp neighbors (route-type) community

Parameters

- *neighbor_addr* location of neighbor.
- ***RTE*** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- ***FILTER*** routing table entries that the command displays. Values include*:*
  - <no parameter> displays all routing table entries. Tabular format.
  - **detail** displays all routing table entries. Data block format.
  - *ipv4_addr* host IPv4 address. Data block format.
  - *ipv4_subnet* subnet address. (CIDR notation). Data block format.
  - *ipv4_subnet* **longer-prefixes** subnet address. (CIDR notation). Tabular format.

Example

- This command displays information for routes advertised to the neighbor at 172.17.254.78

```
switch>show ip bgp neighbors 172.17.254.78 advertised-routes
Route status codes: s - suppressed, * - valid, > - active, e - ECMP

    Network             Next Hop         R Metric  LocPref Path
* >  0.0.0.0/0           -                u 10      4       i (Id 1)
* >  172.31.48.0/23      172.17.254.28    u 0       100     (65533) 65534 i (Id 9)
* >  172.31.50.0/23      172.17.254.28    u 0       100     (65533) 65534 i (Id 10)
* >  172.31.52.0/23      172.17.254.28    u 0       100     (65533) 65534 i (Id 11)
* >  172.31.54.0/23      172.17.254.28    u 0       100     (65533) 65534 i (Id 12)
* >  172.38.254.112/30   172.17.254.28    u 0       100     (65533) 65534 i (Id 13)
* >  172.44.0.34/32      172.17.254.28    u 0       100     (65533) 65534 i (Id 13)
* >  172.71.1.0/24       172.17.254.44    u 0       100     (65533) 65534 i (Id 8)
Rt-ID: 172.31.0.23
switch>
```

show ip bgp neighbors (route-type) community

The **show ip bgp neighbors (route type) community** command displays information for next hop routes to a specified neighbor. Routes are filtered by community.

The **show ip bgp neighbors (route type)** command displays the same information for routes filtered by IP addresses and subnets.

Command Mode

EXEC

Command Syntax

show ip bgp neighbors *n4_addr* ***RTE*** **community** ***CM_1*** **[*CM_2 ...CM_n*][*MATCH*][*INFO*]**

Related Command

show ip bgp neighbors
show ip bgp neighbors (route type)

Parameters

- *n4_addr* location of IPv4 neighbor.
- ***RTE*** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- ***CM_x*** community number or name, as specified in the route map that sets the community list number. The command must list at least one of the following community identifiers:.
  - *aa:nn* AS and network number, separated by colon. Each value ranges from 1 to 65535.
  - *comm_num* community number. Values range from 1 to 4294967040.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- ***MATCH_TYPE*** Routes are filtered based on their communities.
  - <no parameter> routes must match at least one community in the list
  - **exact** route must match all communities and include no other communities.
- ***DATA_OPTION*** Type of information the command displays. Values include:
  - <no parameter> Displays table of of routing entry line items.
  - **detail** Displays data block for each routing table entry.

show ip bgp paths

The **show ip bgp paths** command displays all BGP paths in the database.

Command Mode

EXEC

Command Syntax

```
show ip bgp paths
```

Display Values

- **Refcount**: Number of routes using a listed path.
- **Metric**: The Multi Exit Discriminator (MED) metric for the path.
- **Path**: The autonomous system path for that route, followed by the origin code for that route.

  The MED, also known as the external metric of a route, provides information to external neighbors about the preferred path into an AS with multiple entry points. Lower MED values are preferred.

Example

- This command displays the BGP paths in the switch's database.

```
switch>show ip bgp paths
Refcount Metric     Path
6        0          IGP (Id 1)
2        0          Incomplete (Id 2)
2        0          (100) IGP (Id 5)
switch>
```

show ip bgp peer-group

The **show ip bgp peer-group** command displays the BGP version, address family and group members for all BGP peer groups defined on the switch.

Command Mode

EXEC

Command Syntax

show ip bgp peer-group

Example

- This command displays BGP peer group information for the switch.

```
switch> show ip bgp peer-group
BGP peer-group local
  BGP version 4
  Address family: IPv4 Unicast
  Peer-group members:
  197.254.17.7
  197.254.17.8
BGP peer-group external
  BGP version 4
  Address family: IPv4 Unicast
  Peer-group members:
  121.5.20.21
  121.5.20.25
  121.5.20.31
```

show ip bgp summary

The **show ip bgp summary** command displays BGP path, prefix, and attribute information for all BGP neighbors.

Command Mode

EXEC

Command Syntax

```
show ip bgp summary
```

Display Values

Header Row

- **BGP router identifier**: The router identifier – loopback address or highest IP address.
- **Local AS Number**: AS number assigned to switch

Neighbor Table Columns

- **(First) Neighbor**: IP address of the neighbor.
- **(Second) V**: BGP version number spoken to the neighbor
- **(Third) AS**: Neighbor's Autonomous system number.
- **(Fourth) MsgRcvd:** Number of messages received from the neighbor.
- **(Fifth) MsgSent:** Number of messages sent to the neighbor.
- **(Sixth) InQ:** Number of messages queued to be processed from the neighbor.
- **(Seventh) OutQ:** Number of messages queued to be sent to the neighbor.
- **(Eighth) Up/Down:** Period the BGP session has been in Established state or its current status.
- **(Ninth) State**:State of the BGP session and the number of routes received from a neighbor.

  After the maximum number of routes are received (**maximum paths (BGP)**), the field displays **PfxRcd**, the neighbor is shut down, and the connection is set to Idle.

Example

- This command displays the status of the switch's BGP connections.

```
switch>show ip bgp summary
BGP router identifier 172.26.0.22, local AS number 65533
Neighbor         V  AS      MsgRcvd   MsgSent  InQ OutQ  Up/Down State/PfxRcd
172.17.254.78    4  65534       187       191    0    0 02:49:40 7
172.17.254.2     4  65533       184       191    0    0 02:59:41 7
switch>
```

show ip community-list

The **show ip community-list** command displays the BGP community lists configured on the switch.

Command Mode

EXEC

Command Syntax

show ip community-list [*COMMUNITY_LIST*]

Parameters

- ***COMMUNITY_LIST*** community list for which command displays information.
  - <no parameter> command displays information for all community lists.
  - *listname* name of the community list (text string).

Example

- This command displays the BGP paths in the switch's database.

```
switch>show ip community-list hs-comm-list
ip community-list standard hs-comm-list permit 0:10
switch>
```

show ip extcommunity-list

The **show ip extcommunity-list** command displays the BGP extended community lists configured on the switch.

Command Mode

EXEC

Command Syntax

show ip extcommunity-list [*COMMUNITY_LIST*]

Parameters

- ***COMMUNITY_LIST*** extended community list for which command displays information.
  - <no parameter> command displays information for all extended community lists.
  - *listname* name of the extended community list (text string).

Example

- This command displays the extended extcommunity lists on the switch.

```
switch>show ip extcommunity-list
ip extcommunity-list standard hs-extcomm-list permit rt 3050:20
ip extcommunity-list standard hs-extcomm-list permit soo 172.17.52.2:30
ip extcommunity-list standard hs-extcomm-list permit rt 3050:70000
switch>
```

show ipv6 bgp

The **show ipv6 bgp** command displays IPv6 Border Gateway Protocol (BGP) routing table entries. The output format depends on the command parameters:

The output format depends on the command parameters:

- Data block format displays comprehensive information for each specified BGP routing table entry.
- Tabular format displays routing table entries in tabular format for the specified IP addresses.

Command Mode

EXEC

Command Syntax

show ipv6 bgp [***FILTER***]

Parameters

- ***FILTER*** routing table entries that the command displays. Values include*:*
  - — <no parameter> displays all routing table entries. Tabular format.
  - — **detail** displays all routing table entries. Data block format.
  - — *ipv6_addr* IPv6 host address. Data block format.
  - — *ipv6_prefix* IPv6 prefix address. (CIDR notation). Data block format.
  - — *ipv6_prefix* **longer-prefixes** IPv6 prefix address. (CIDR notation). Tabular format.

Examples

- This command displays the routing data blocks for a specified IPv6 prefix.

```
switch>show ipv6 bgp fd7a:9924:5804:1134::/64 longer-prefixes
Route status codes: s - suppressed, * - valid, > - active, e - ECMP

     Network             Next Hop         R Metric  LocPref Path
* >  fd7a:9924:5804:1134::/64 -                u 0       0       ? (Id 1)
*    fd7a:9924:5804:1134::/64 fd7a:9924:5804:fe4c::1 u 0       100     (65533)
65534 i (Id 597)
```

- This command displays the routing table for a specified IPv6 prefix.

```
switch>show ipv6 bgp fd7a:9924:5804:1134::/64
BGP routing table entry for fd7a:9924:5804:1134::/64
 Paths: 2 available
  Local
    - from - (172.26.0.22)
      Origin INCOMPLETE, metric 0, localpref 0, valid, local, best
  (65533) 65534
    fd7a:9924:5804:fe4c::1 from fd7a:9924:5804:fe4c::1 (172.26.0.34)
      Origin IGP, metric 0, localpref 100, valid, external
switch>
```

show ipv6 bgp community

The **show ipv6 bgp community** command displays IPv6 Border Gateway Protocol (BGP) routing table entries, filtered by community.

Command Mode

EXEC

Command Syntax

```
show ipv6 bgp community [COMM_1 ... COMM_n][MATCH_TYPE][DATA_OPTION]
```

Parameters

- *COMM_x* community number or name, as specified in the route map that sets the community list number.
  - *aa:nn* AS and network number, separated by colon. Each value ranges from 1 to 65535.
  - *comm_num* community number. Values range from 1 to 4294967040.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- *MATCH_TYPE* Routes are filtered based on their communities.
  - <no parameter> routes must match at least one community in the list
  - **exact** route must match all communities and include no other communities.
- *DATA_OPTION* Type of information the command displays. Values include:
  - <no parameter> Displays table of of routing entry line items.
  - **detail** Displays data block for each routing table entry.

show ipv6 bgp neighbors

The **show ipv6 bgp neighbors** command displays IPv6 Border Gateway Protocol (BGP) and TCP session data for a specified neighbor. Command displays data for all neighbors if an address is not included.

Command Mode

EXEC

Command Syntax

show ipv6 bgp neighbor [***NEIGHBOR_ADDR***]

Parameters

- ***NEIGHBOR_ADDR*** location of neighbors. Options include:
  - <no parameter> command displays information for all neighbors.
  - *ipv6_addr* command displays information for specified neighbor.

Example

- This command displays information for the neighbor at fd7a:629f:52a4:fe01::2

```
switch>show ipv6 bgp neighbors fd7a:629f:52a4:fe01::2
BGP neighbor is fd7a:629f:52a4:fe01::2, remote AS 65533
Description: v6-bgp-to-magensium
BGP version is 4, remote router ID 172.26.0.23
Negotiated version is 4
TTL is 0
holdtime is 180
restart-time is 0
Restarting: no
Current state is Established
Updates received: 256
Updates sent: 4787
Total messages received: 11097
Total messages sent: 15250
Last state was OpenConfirm
Last event was RecvKeepAlive
Last error code was 0
Last error subcode was 0
Established time: 652492 seconds
Number of transitions to established: 1
Local TCP address is fd7a:629f:52a4:fe01::1
Local AS is 65533
Local router ID is 172.26.0.22
Capabilities                     Snt   Rcv   Neg
-----------------------------------------------
Multiprotocol IPv4 Unicast       yes   yes   yes
Graceful Restart IPv4 Unicast    no    no    no
Multiprotocol IPv4 Multicast     no    no    no
Graceful Restart IPv4 Multicast  no    no    no
Multiprotocol IPv6 Unicast       yes   yes   yes
Graceful Restart IPv6 Unicast    no    no    no
Multiprotocol IPv4 VPN           no    no    no
Graceful Restart IPv4 VPN        no    no    no
Route Refresh                    yes   yes   yes
Send End-of-RIB messages         no    no    no
Dynamic Capabilities             no    no    no
```

show ipv6 bgp neighbors (route type)

The **show ipv6 bgp neighbors (route type)** command displays information for next hop routes to a specified IPv6 BGP neighbor. The **show ipv6 bgp neighbors (route type) community** command displays the same information for routes filtered by communities.

The output format depends on the selected ***FILTER*** parameter:

- Data block format displays comprehensive information for each specified route.
- Tabular format displays routing table entries in tabular format for the specified IP addresses.

Commands that do not include a route type revert to the **show ipv6 bgp neighbors** command.

Command Mode

EXEC

Command Syntax

show ipv6 bgp neighbors *neighbor_addr* ***ROUTE_TYPE*** [***FILTER***]

Parameters

- *neighbor_addr* location of neighbor.
- ***ROUTE_TYPE*** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- ***FILTER*** routing table entries that the command displays. Options include*:*
  - <no parameter> displays all routing table entries. Tabular format.
  - **detail** displays all routing table entries. Data block format.
  - *ipv6_addr* IPv6 host address. Data block format.
  - *ipv6_prefix* IPv6 prefix address (CIDR notation). Data block format..
  - *ipv6_prefix* **longer-prefixes** IPv6 prefix address. (CIDR notation). Tabular format.

Example

- This command displays information for routes advertised to the neighbor at fd7a:629f:52a4:1::/64

```
switch>show ipv6 bgp neighbors fd7a:629f:52a4:fe01::2 routes fd7a:629f:52a4::/48
longer-prefixes
Route status codes: s - suppressed, * - valid, > - active, e - ECMP

     Network             Next Hop         R Metric  LocPref Path
*    fd7a:629f:52a4:1::/64 fd7a:629f:52a4:fe61::2 u 0       100     (65533) ? (Id
7) Rt-ID: 172.26.0.23
*    fd7a:629f:52a4:1001::/64 fd7a:629f:52a4:fe61::2 u 0       100     (65533) ?
(Id 7) Rt-ID: 172.26.0.23
*    fd7a:629f:52a4:1616::/64 fd7a:629f:52a4:1900:400::1 u 0       100     (65533)
? (Id 9) Rt-ID: 172.26.0.23
*    fd7a:629f:52a4:1a00::23/128 fd7a:629f:52a4:fe50::2 u 0       100     (65533)
? (Id 11) Rt-ID: 172.26.0.23
* >  fd7a:629f:52a4:fe70::/64 fd7a:629f:52a4:fe50::2 u 0       100     (65533)
65534 i (Id 59) Rt-ID: 172.26.0.23
*    fd7a:629f:52a4:fee4::/62 fd7a:629f:52a4:fe08::3 u 0       100     (65533) ?
(Id 24) Rt-ID: 172.26.0.23
switch>
```

show ipv6 bgp neighbors (route type) community

The **show ipv6 bgp neighbors (route type) community** command displays information for next hop routes to a specified IPv6 BGP neighbor. Routes are filtered by community.

The **show ipv6 bgp neighbors (route type)** command displays the same information for routes filtered by IP addresses and prefixes.

Command Mode

EXEC

Command Syntax

show ipv6 bgp neighbors n6_addr ***RTE*** **community** ***CM_1*** **[*CM_2 ..CM_n*][*MATCH*][*INFO*]**

Parameters

- *n6_addr* location of IPv6 neighbor.
- ***RTE*** type of route that the command displays. Options include:
  - **advertised-routes** displays routes advertised to the specified neighbor.
  - **received-routes** displays routes received from specified neighbor (accepted and rejected).
  - **routes** displays routes received and accepted from specified neighbor.
- ***CM_x*** community number or name, as specified in the route map that sets the community list number. The command must list at least one of the following community identifiers:.
  - *aa:nn* AS and network number, separated by colon. Each value ranges from 1 to 65535.
  - *comm_num* community number. Values range from 1 to 4294967040.
  - **internet** advertises route to Internet community.
  - **local-as** advertises route only to local peers.
  - **no-advertise** does not advertise route to any peer.
  - **no-export** advertises route only within BGP AS boundary.
- ***MATCH*** Routes are filtered based on their communities.
  - <no parameter> routes must match at least one community in the list
  - **exact** route must match all communities and include no other communities.
- ***INFO*** Type of information the command displays. Values include:
  - <no parameter> Displays table of of routing entry line items.
  - **detail** Displays data block for each routing table entry.

Example

- This command displays the BGP routes in the fd7a:718a:523c:fe4c::1 network that are assigned the community of 65533:100.

```
switch>show ipv6 bgp neighbors fd7a:718a:523c:fe4c::1 advertised-routes community
65533:100
Route status codes: s - suppressed, * - valid, > - active, e - ECMP

     Network             Next Hop         R Metric  LocPref Path
* >  172.17.254.0/30     172.17.254.1     u 1       0       65533 i (Id 0)
switch>
```

show ipv6 bgp summary

The **show ipv6 bgp summary** command displays BGP path, prefix, and attribute information for all BGP IPv6 neighbors.

Command Mode

EXEC

Command Syntax

```
show ipv6 bgp summary
```

Display Values

Header Row

- **BGP router identifier**: The router identifier – loopback address or highest IP address.
- **Local AS Number**: AS number assigned to switch

Neighbor Table Columns

- **(First) Neighbor**: IP address of the neighbor.
- **(Second) V**: BGP version number spoken to the neighbor
- **(Third) AS**: Neighbor's Autonomous system number.
- **(Fourth) MsgRcvd:** Number of messages received from the neighbor.
- **(Fifth) MsgSent:** Number of messages sent to the neighbor.
- **(Sixth) InQ:** Number of messages queued to be processed from the neighbor.
- **(Seventh) OutQ:** Number of messages queued to be sent to the neighbor.
- **(Eighth) Up/Down:** Period the BGP session has been in Established state or its current status.
- **(Ninth) State**:State of the BGP session and the number of routes received from a neighbor.

  After the maximum number of routes are received (**maximum paths (BGP)**), the field displays **PfxRcd**, the neighbor is shut down, and the connection is set to Idle.

Example

- This command displays the status of the switch's BGP connections.

```
switch>show ipv6 bgp summary
BGP router identifier 172.26.0.22, local AS number 65533
Neighbor         V  AS      MsgRcvd   MsgSent  InQ OutQ  Up/Down State  PfxRcd
fd7a:629f:52a4:fe4c::1 4  65534      6030      6029    0    0    2d13h Estab  8
fd7a:629f:52a4:fe01::2 4  65533      6212      6294    0    0    3d08h Estab  818
switch>
```

shutdown (BGP)

The **shutdown** command disables BGP on the switch without modifying the BGP configuration.

The **no shutdown** and **default shutdown** commands remove the shutdown command from the configuration, re-enabling the BGP instance.

Command Mode

Router-BGP Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Examples

- This command disables BGP on the switch.

```
switch(config)#router bgp 9
switch(config-router-bgp)#shutdown
switch(config-router-bgp)#
```

- This command enables BGP on the switch.

```
switch(config)#router bgp 9
switch(config-router-bgp)#no shutdown
switch(config-router-bgp)#
```

timers bgp

The **timers bgp** command configures the BGP keepalive and hold times.Timer settings apply to each peer connection. The **neighbor timers** command configures the times on a specified peer connection.

- Keepalive time: period between the transmission of consecutive keepalive messages.
- Hold time: period the switch waits for a keepalive or UPDATE message before it disables peering.

The hold time must be at least 3 seconds and should be three times longer than the keepalive setting.

The **no timers bgp** and **default timers bgp** commands remove the **timers bgp** command from the configuration, which returns the time settings to their defaults

- keepalive: 60 seconds
- hold time: 180 seconds

Command Mode

Router-BGP Configuration

Command Syntax

```
timers bgp keep_alive hold_time
no timers bgp
default timers bgp
```

Parameters

- *keep_alive* keepalive period, in seconds. Values include
  - **0** keepalive messages are not sent
  - **1** to **3600** keepalive time (seconds).
- *hold_time* hold time. Values include
  - **0** peering is not disabled by timeout expiry; keepalive packets are not sent.
  - **3** to **7200** hold time (seconds).

Example

- This command sets the keepalive time to 30 seconds and the hold time to 90 seconds.

```
switch(config)#router bgp 9
switch(config-router-bgp)#timers bgp 30 90
switch(config-router-bgp)#
```

Chapter 22

Routing Information Protocol

Routing Information Protocol (RIP) is a distance-vector routing protocol typically used as an interior gateway protocol (IGP). Arista switches supports RIP version 2, which is defined by RFC 2453.

This chapter contains the following sections.

- Section 22.1: RIP Conceptual Overview
- Section 22.2: Running RIP on the Switch
- Section 22.3: RIP Commands

22.1 RIP Conceptual Overview

Routing Information Protocol (RIP) is a distance-vector routing protocol typically used as an interior gateway protocol (IGP). RIP uses only hop count to determine the shortest path to a destination. To avoid loops, RIP limits its paths to a maximum of 15 hops, making it an ineffective protocol for large networks. RIP Version 2 supports Classless Inter-Domain Routing (CIDR) and uses IP multicasting at address 224.0.0.9 to share the routing table with adjacent routers.

RIP sends routing-update messages at regular intervals and when the network topology changes. When a switch receives a routing update that includes changes to an entry, it updates its routing table to reflect the new route. Because RIP transmits the entire routing table every 30 seconds, RIP updates can generate heavy traffic loads in large or complicated networks.

Each switch also sends a list of distance-vectors to each of its neighbors periodically. The distance-vector is the metric RIP uses to express the cost of a route, and it describes the number of hops required to reach a destination. Each hop is typically assigned a hop count value of 1, and the router adds 1 to the metric when it receives a routing update and adds the network to its routing table.

To remove dead routes from its routing table, RIP marks a route for deletion if the router does not receive an advertisement for it within the expiration interval, then removes it from the routing table after the deletion interval.

22.2 Running RIP on the Switch

22.2.1 Accessing RIP Configuration Mode and Enabling RIP

22.2.1.1 RIP Configuration Mode

The **router rip** command places the switch in router-RIP configuration mode to configure the Routing Information Protocol (RIP) routing process.

Example

- This command places the switch in router-rip configuration mode.

```
switch(config)#router rip
switch(config-router-rip)#
```

Using the **router rip** command puts the switch in router-RIP configuration mode, but does not enable RIP on the switch.

22.2.1.2 Enabling RIP

Routing Information Protocol (RIP) is disabled on the switch by default. To enable RIP, use the **no** form of the **shutdown (RIP)** command in router-RIP configuration mode.

Example

- This command enables RIP on the switch.

```
switch(config-router-rip)#no shutdown
switch(config-router-rip)#
```

Issuing this command enables RIP, but to send and receive RIP route updates and to route packets via RIP you must also specify interfaces on which RIP will run by using the **network (RIP)** command.

22.2.1.3 Disabling RIP

You can disable RIP in two ways. The **shutdown (RIP)** command disabled RIP on the switch but leaves all user-entered router-RIP configuration statements in *running-config*. The **no** form of the **router rip** command disables RIP and removes all user-entered router-rip configuration statements from *running-config*.

Examples

- This command disables RIP on the switch and removes all user-entered router-RIP configuration.

```
switch(config)#no router rip
switch(config)#
```

- This command disables RIP on the switch, but preserves all user-entered router-RIP configuration.

```
switch(config-router-rip)#shutdown
switch(config-router-rip)#
```

22.2.2 Configuring RIP

Issuing the **no** form of the **shutdown (RIP)** command in router-RIP configuration mode enables RIP, but to run RIP on an interface you must specify a RIP network by using the **network (RIP)** command.

You can also configure the redistribution of routes learned from other protocols, set the default metric and administrative distance for redistributed routes, configure the timing of various RIP events, and configure specific interfaces to send RIP update packets by broadcast instead of multicast.

22.2.2.1 Specifying RIP Networks

The **network (RIP)** command identifies networks on which RIP will run and also specifies which routes RIP will accept into its routing table. You can issue the command multiple times to build up a list of RIP networks. No RIP networks are configured by default, so in order to route packets and send and receive RIP updates you must specify one or more RIP networks.

To disable RIP on a specific network, use the **no network (RIP)** command.

Examples

- This command enables RIP on 192.168.1.1/24

```
switch(config-router-rip)#network 192.168.1.1/24
switch(config-router-rip)#
```

- This command disables RIP on 192.168.1.1/24

```
switch(config-router-rip)#no network 192.168.1.1/24
switch(config-router-rip)#
```

22.2.2.2 Redistributing Routes Learned from Other Protocols into RIP

To enable route import from a specified protocol into RIP, use the **redistribute (RIP)** command. You can also apply a route map to the incoming routes to filter which routes are added to the RIP routing table. All connected routes are redistributed into RIP by default; you can filter them by using the **redistribute** command and specifying a route map.

Example

- This command redistributes all routes learned from OSPF into RIP.

```
switch(config-router-rip)#redistribute OSPF
switch(config-router-rip)#
```

22.2.2.3 Configuring RIP Timers

When RIP is running on the switch, it sends unsolicited route updates and deletes expired routes at regular intervals. To configure the timing of those events, use the **timers basic (RIP)** command. The command takes three parameters: the update interval, the route expiration time, and the route deletion time.

The update interval is the time in seconds that the switch waits between sending unsolicited RIP route updates to its neighbors. The route expiration time is how long the switch waits before marking an unadvertised route for deletion (the counter resets whenever an advertisement for the route is received). And the route deletion time is how long the switch waits between marking a route for deletion and removing it from the routing table. During the deletion interval, the switch continues to forward packets on the route.

Example

- This command sets the update interval to 60 seconds, expiration time to 90 seconds, and deletion time to 150.

```
switch(config-router-rip)#timers basic 60 90 150
switch(config-router-rip)#
```

22.2.2.4 Configuring an Interface to Transmit Broadcast RIP Updates

By default, the switch uses RIP version 2 and multicasts RIP update packets from all participating interfaces. To reconfigure a specific interface to send updates as broadcast packets rather than multicast packets, use the **ip rip v2-broadcast** command in the configuration mode for the interface.

Example

- The following commands configure RIP version 2 broadcasting on interface Ethernet 5.

```
switch(config)#interface ethernet5
switch(config-if-Et5)#ip rip v2-broadcast
switch(config-if-Et5)#exit
switch(config)#
```

22.2.3 Displaying RIP Information

22.2.3.1 Displaying RIP Routes

To see a listing of the RIP routes in the switch's routing table, use the **show ip rip database** command. (You can also display similar information using the RIP option in the **show ip route** command.)

Examples

- This command displays all active rip routes.

```
switch>show ip rip database
192.168.11.0/24 directly connected, Et0
192.168.13.0/24
[1] via 192.168.14.2, 00:00:25, Et0
[2] via 192.168.15.2, 00:00:20, Et1
182.168.13.0/24
[1] via 182.168.14.2, 00:00:25, Et3
```

- This command submits a query for RIP route information for a network..

```
switch>show ip rip database 192.168.13.0/16
192.168.13.0/24
[1] via 192.168.14.2, 00:00:25, Et0
[2] via 192.168.15.2, 00:00:20, Et1
```

22.2.3.2 Displaying RIP Route Gateways

To see information about the switch's RIP route gateways, use the **show ip rip neighbors** command. The output displays the IPv4 address, the last heard time of the gateway, and characteristic flags applying to the gateway.

Example

- This command displays information about all the gateways of RIP routes..

```
switch>show ip rip neighbors
Gateway     Last-Heard      Bad-Packets     Bad-Routes    Flags
10.2.12.33  00:00:15                                      SRC, TRSTED,
                                                          ACCPTED, RJCTED,
                                                          Q_RJCTED, AUTHFAIL
```

22.3 RIP Commands

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Commands

- router rip . . . . . Page 985

Interface Configuration Commands

- ip rip v2-broadcast . . . . . Page 982

Router-RIP Configuration Mode

- default-metric . . . . . Page 980
- distance (RIP) . . . . . Page 981
- network (RIP) . . . . . Page 983
- redistribute (RIP) . . . . . Page 984
- shutdown (RIP) . . . . . Page 988
- timers basic (RIP) . . . . . Page 989

Display Commands – EXEC Mode

- show ip rip database . . . . . Page 986
- show ip rip neighbors . . . . . Page 987

default-metric

The **default-metric** command specifies the metric value assigned to RIP routes learned from other protocols. All routes imported into RIP receive the default metric unless a matching route-map exists for the route. The route metric of 0 is assigned to redistributed connected and static routes. Default-metric values range from 0 to 16 with a default value of 1.

The **no default-metric** and **default default-metric** commands remove the **default-metric** command from ***running-config*** and returns the default-metric value to its default value of 1.

Command Mode

Router-RIP Configuration

Command Syntax

```
default-metric metric_value
no default-metric
default default-metric
```

Parameters

- *metric_value* default metric value assigned. Values range from 0 to 16; default is 1.

Example

- This command sets the default metric value to five.

```
switch(config)#router rip
switch(config-router-rip)#default-metric 5
switch(config-router-rip)#
```

distance (RIP)

The **distance** command assigns an administrative distance to routes that the switch learns through RIP. Routers use administrative distances to select a route when two protocols provide routing information to the same destination. Distance values range from 1 to 255; lower distance values correspond to higher reliability. The default RIP distance value is 120.

The **no distance** and **default distance** commands restore the administrative distance default value of 120 by removing the **distance** command from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

distance *distance_value*
no distance
default distance

Parameters

- *distance_value* distance assigned to RIP routes. Values range from 1 to 255.

Examples

- These commands assign an administrative distance of 75 to RIP routes.

```
switch(config)#router rip
switch(config-router-rip)#distance 75
switch(config-router-rip)#
```

ip rip v2-broadcast

The **ip rip v2-broadcast** command specifies the transmission of Routing Information Protocol (RIP) Version 2 update packets from the configuration mode interface as broadcast packets instead of multicast packets. Requests and responses are sent to the IP broadcast address 255.255.255.255 instead of the IP multicast address 224.0.0.9. If the interface is not multicast capable, then updates are broadcast.

The **no rip v2-broadcast** and **default rip v2-broadcast commands** specify the transmission of RIP v2 as multicast if the configuration mode interface is multicast capable to the reserved multicast address, 224.0.0.9). Updates are broadcast if the interface is not multicast capable.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip rip v2-broadcast
no ip rip v2-broadcast
default ip rip v2-broadcast
```

Example

- The following example configures version 2 broadcasting on interface Ethernet 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#ip rip v2-broadcast
switch(config-if-Et5)#exit
switch(config)#
```

network (RIP)

The **network** command specifies a network on which the switch runs Routing Information Protocol (RIP), and also specifies which routes will be accepted into the RIP routing table. Multiple network commands can be issued to create a network list on which RIP runs.

The switch enables RIP on all interfaces in the specified network.

The **no network** and **default network** commands disable RIP on the specified network by removing the corresponding **network** command from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

```
network NETWORK_ADDRESS
no network NETWORK_ADDRESS
default network NETWORK_ADDRESS
```

Parameters

- *NETWORK_ADDRESS* network IP address. Entry formats include the following:
  - *ipv4_subnet* IPv4 subnet (CIDR notation).
  - *ipv4_addr* **mask** *wildcard_mask* IP address and wildcard-mask.

Examples

- This command enables RIP on 192.168.1.1/24

```
switch(config)#router rip
switch(config-router-rip)#network 192.168.1.1/24
switch(config-router-rip)#
```

- This command also enables RIP on 192.168.1.1/24

```
switch(config-router-rip)#network 192.168.1.1 mask 0.0.0.255
switch(config-router-rip)#
```

redistribute (RIP)

The **redistribute** command enables the importing of routes from a specified routing domain to RIP.

- **connected** by default, RIP redistributes all connected routes that are established when IP is enabled on an interface. The route-map parameter facilitates the exclusion of connected routes from redistribution by specifying a route map that denies the excluded routes.
- **BGP**, **OSPF**, and **IP static routes** by default, routes are not redistributed. The redistribution command without the route-map parameter faciltates the redistribution of all routes from the specified source.

The **no redistribute** and **default redistribute** commands reset the default route redistribution setting by removing the **redistribute** statement from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

```
redistribute connected ROUTE_MAP
redistribute ROUTE_TYPE [ROUTE_MAP]
no redistribute connected ROUTE_MAP
no redistribute ROUTE_TYPE
default redistribute connected ROUTE_MAP
default redistribute ROUTE_TYPE
```

Parameters

- *ROUTE_TYPE* source from which routes are redistributed. Options include:
  - **aggregate** BGP aggregate routes.
  - **BGP** routes from a BGP domain.
  - **OSPF** routes from an OSPF domain.
  - **OSPF match external** Routes external to RIP, but imported from OSPF.
  - **OSPF match internal** OSPF routes that are internal to the AS.
  - **static** IP static routes.
- *ROUTE_MAP* route map that determines the routes that are redistributed. Options include:
  - <no parameter> all routes are redistributed.
  - **route-map** *map_name* only routes in the specified route map are redistributed.

Example

- These commands redistribute OSPF routes into RIP.

```
switch(config)#router rip
switch(config-router-rip)#redistribute OSPF
switch(config-router-rip)#
```

router rip

The **router rip** command places the switch in router-rip configuration mode to configure the Routing Information Protocol (RIP) routing process. Router-rip configuration mode is not a group change mode; *running-config* is changed immediately upon command entry. The **exit** command does not affect *running-config*.

The **no router rip** and **default router rip** commands disable RIP and remove all user-entered *router-rip* configuration statements from *running-config*. To disable RIP without removing configuration statements, use the **shutdown (RIP)** command.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
router rip
no router rip
default router rip
```

Commands Available in router-rip Configuration Mode

- **default-metric**
- **distance (RIP)**
- **network (RIP)**
- **redistribute (RIP)**
- **shutdown (RIP)**
- **timers basic (RIP)**

Example

- This command places the switch in router-rip configuration mode.

```
switch(config)#router rip
switch(config-router-rip)#
```

show ip rip database

The **show ip rip database** command displays information about routes in the Routing Information Base. The default command displays active routes and learned routes not used in deference to higher priority routes from other protocols.

This query has the following forms:

- default (no arguments): information about all RIP routes.
- IPv4 address and mask: information about the referenced addresses
- **active:** information about routes not superceded by routes from other protocols.

Command Mode

EXEC

Command Syntax

```
show ip rip database [FILTER]
```

Parameters

- *FILTER* routing table entries that the command displays. Values include*:*
  - — <no parameter> displays all routing table entries
  - — **active** displays all active routing table entries.
  - — *net_addr* subnet address. (CIDR or address-mask). Command displays entries in this subnet.

Examples

- This command displays all active rip routes.

```
switch>show ip rip database active
192.168.11.0/24 directly connected, Et0
192.168.13.0/24
[1] via 192.168.14.2, 00:00:25, Et0
[2] via 192.168.15.2, 00:00:20, Et1
182.168.13.0/24
[1] via 182.168.14.2, 00:00:25, Et3
```

- This command submits a query for RIP route information for a network.

```
switch>show ip rip database 192.168.13.0/16
192.168.13.0/24
[1] via 192.168.14.2, 00:00:25, Et0
[2] via 192.168.15.2, 00:00:20, Et1
```

- This command returns information for all RIP routes.

```
switch>show ip rip database
223.1.0.0/255.255.255.0
[1] via 10.8.31.15, 00:00:21, Et2, holddown
223.2.0.0/255.255.255.0
[1] via 10.8.31.15, 00:00:21, Et2, holddown
223.3.0.0/255.255.255.0
[1] via 10.8.31.15, 00:00:21, Et2, inactive
223.212.0.0/255.255.255.0
[1] via 10.8.31.15, 00:00:21, Et2, active
223.214.0.0/255.255.255.0
[1] via 10.8.12.17, 00:00:30, Et0, active
```

show ip rip neighbors

The **show ip rip neighbors** command displays information about all RIP route gateways. The output displays the IPv4 address, the last heard time of the gateway, and characteristic flags applying to the gateway.

Command Mode

EXEC

Command Syntax

show ip rip neighbors

Examples

- The **show ip rip neighbors** query displays information about all the gateways of RIP routes.

```
switch>show ip rip neighbors
Gateway     Last-Heard      Bad-Packets     Bad-Routes    Flags
10.2.12.33  00:00:15                                      SRC, TRSTED,
                                                          ACCPTED, RJCTED,
                                                          Q_RJCTED, AUTHFAIL
```

shutdown (RIP)

The **shutdown** command disables RIP on the switch without modifying the RIP configuration. RIP is disabled by default.

The **no shutdown** command enables RIP. The **default shutdown** command disables RIP.

Command Mode

Router-RIP Configuration

Command Syntax

```
shutdown
no shutdown
default shutdown
```

Examples

- This command disables RIP on the switch.

```
switch(config)#router rip
switch(config-router-rip)#shutdown
switch(config-router-rip)#
```

- This command enables RIP on the switch.

```
switch(config-router-rip)#no shutdown
switch(config-router-rip)#
```

timers basic (RIP)

The **timers basic** command configures the update interval, the expiration time, and the deletion time for routes received and sent through RIP. The command requires value declaration of all values.

- The update time is the interval between unsolicited route responses. The default is 30 seconds.
- The expiration time is initialized when a route is established and any time an update is received for the route. If the specified period elapses from the last time the route update was received, then the route is marked as inaccessible and advertised as unreachable. However, the route forwards packets until the deletion time expires. The default value is 180 seconds.
- The deletion time is initialized when the expiration time has elapsed. On initialization of the deletion time, the route is no longer valid; however, it is retained in the routing table for a short time so that neighbors can be notified that the route has been dropped. Upon expiration of the deletion time, the route is removed from the routing table. The default is 120 seconds.

The **no timers basic** and **default timers basic** commands return the timer values to their default values by removing the **timers-basic** command from ***running-config***.

Command Mode

Router-RIP Configuration

Command Syntax

```
timers basic update_time expire_time deletion_time
no timers basic
default timers basic
```

Parameters

- *update_time* rate at which updates are sent.
- *expire_time* period a route is valid after it is established or updated. Must be greater than *update_time*.
- *deletion_time* interval after expiration when route is removed from routing table.

Parameter values are in seconds and range from 5 to 2,147,483,647.

Example

- This command sets the update time to 60 seconds, expiration time to 90 seconds, and deletion time to 150.

```
switch(config)#router rip
switch(config-router-rip)#timers basic 60 90 150
switch(config-router-rip)#
```

Chapter 23

Multicast

IP multicast is the transmission of data packets to multiple hosts through a common IP address. Arista switches support multicast transmissions through IGMP, IGMP Snooping, and PIM-SM.

These sections describe the Arista multicast implementation.

- Section 23.1: Introduction is a chapter overview and lists the features supported by Arista switches.
- Section 23.2: Multicast Architecture describes multicast data structures
- Section 23.3: Multicast Protocols describes the multicast protocols – IGMP and PIM.
- Section 23.4: Configuring Multicast describes configuration tasks that implement multicast.
- Section 23.5: Multicast Example provides a multicast implementation scenario.
- Section 23.6: Multicast Commands contains multicast command descriptions.
- Section 23.7: IGMP Commands contains IGMP command descriptions.
- Section 23.8: IGMP Snooping Commands contains IGMP Snooping command descriptions.
- Section 23.9: PIM Commands contains PIM command descriptions.

23.1 Introduction

Arista switches provide layer 2 multicast filtering and layer 3 routing features for applications requiring IP multicast services. The switches support over a thousand separate routed multicast sessions at wire speed without compromising other Layer 2/3 switching features. Arista switches support IGMP, IGMP snooping, and PIM-SM to simplify and scale data center multicast deployments.

23.1.1 Supported Features

Arista switches support these multicast functions:

- IGMPv2 router-side functionality
- IGMPv3
- IGMPv2 Snooping based on mac address filtering
- PIM functions:
  - PIM-SM v2 basic functionality
  - Register encapsulation when the DR
  - Register Decapsulation when the RP
  - Data-triggered PIM asserts
  - Static RP configuration
  - Anycast RP

— Dynamic RPs through Bootstrap Routers (BSRs)
— Flooding in each egress VLAN constrained by IGMP snooping
— Multicast routing to/from MLAGs in limited scenarios.

Multicast and unicast use the same routing table. Unicast routes use TCAM resources, which may also impact the maximum number of multicast routes.

Table 23-1 lists the multicast features that each Arista switch platform supports.

Feature	7100 Series	7500 Series	7048	7050 Series
IGMPv2 Snooping	YES	YES	YES	YES
IGMPv2 Querier	YES	YES	YES	YES
IGMPv3 Snooping	YES	YES	YES	YES
PIM-SM + IGMP	YES	YES	YES	YES
PIM-MBR	YES	YES	YES	YES
Anycast RP	YES	YES	YES	YES

Table 23-1 Multicast Feature Support

23.1.2 Features Not Supported

These multicast functions are not supported by Arista switches:

- Multicast Functionality
  — (*,*,G) forwarding or boundary routers
  — Multicast MIBs
  — Router applications joining multicast groups
- PIM Functionality
  — PIMv1 support
  — PIM-DM

23.2 Multicast Architecture

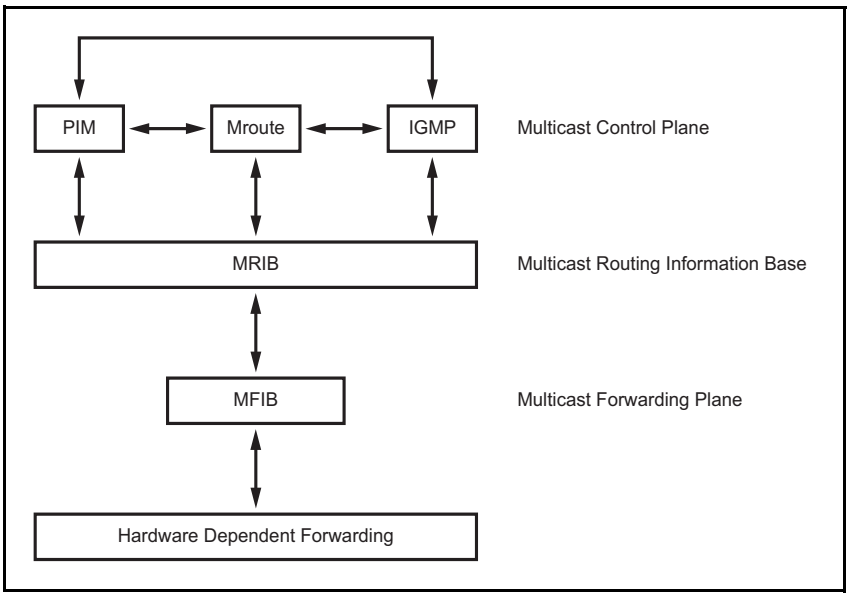
IP multicast is data transmission to a subset of all hosts through a single multicast group address. Multicast packets are delivered using best-effort reliability, similar to unicast packets. Senders use the multicast address as the destination address. Any host, regardless of group membership, can send to a group. However, only group members receive messages sent to a group address.

IP multicast addresses range from 224.0.0.0 to 239.255.255.255. Multicast routing protocol control traffic reserves the address range 224.0.0.0 to 224.0.0.255. The address 224.0.0.0 is never assigned to any group.

Multicast group membership is dynamic; hosts join and leave at any time. There is no restriction on the location or number of members in a group. A host can simultaneously belong to multiple multicast groups. A group's activity level and membership can vary over time.

Figure 23-1 depicts the components that comprise the multicast architecture. This section describes multicast components depicted in the figure.

Figure 23-1 Multicast Architecture



23.2.1 Multicast Control Plane

The Multicast Control Plane builds and maintains multicast distribution trees. It consists of PIM, IGMP, and the mroute table. Mroute table changes, additions, and deletions are learned through PIM or IGMP, communicated across the MRIB, and distributed to MFIB for multicast forwarding. Packet reception events that require control plane updates are handled between MRIB and MFIB.

- Protocol Independent Multicast (PIM) builds and maintains multicast routing trees using reverse path forwarding (RPF) on a unicast routing table.
- Internet Group Management Protocol (IGMP) identifies multicast group members on subnets directly connected to the switch. Hosts manage multicast group membership with IGMP messages.
- The switch maintains a mroute (multicast routing) table when running PIM to provide forwarding tables used to deliver multicast packets.

  The mroute table stores the states of inbound and outbound interfaces for each source-group pair (S,G). The switch discards and forwards packets on the basis of this state information. Each table entry, referred to as an mroute, corresponds to a unique (S,G) and contains:

— the multicast group address
— the multicast source address (or * for all sources)
— the inbound interface
— a list of outbound interfaces

23.2.2 Multicast Forwarding Plane

The Multicast Forwarding Plane consists of the Multicast Forwarding Information Base (MFIB), a forwarding engine that is independent of multicast routing protocols. MFIB responsibilities include:

- Forwarding multicast packets.
- Registering with the MRIB to learn the entry and interface flags set by the control plane.
- Handling data-driven events that the control plane requires.
- Maintaining statistics about received, dropped, and forwarded multicast packets.

MFIB refines multicast routes created by PIM and IGMP into a protocol-independent format for hardware packet forwarding. Each MFIB table entry consists of an (S,G) or (*,G) route, an input RPF VLAN, and a list of Layer 3 output interfaces. MFIB uses platform-dependent management software to load multicast routing information to the hardware FIB and hardware multicast expansion table (MET).

MFIB uses a core forwarding engine for interrupt-level (fast switching) and process-level (process switching) forwarding. MFIB fast-switches inbound multicast packets that match an MFIB forwarding entry and process-switches packets requiring a forwarding entry if a matching entry does not exist.

23.2.3 Multicast Routing Information Base (MRIB)

The MRIB is the channel between Multicast Control Plane clients and the Multicast Forwarding Plane. The **show ip mroute** displays MRIB entries as (*, G), (S, G), and (*, G/m) multicast entries.

MRIB entries are based on source, group, and group masks. The entries are associated with a list of interfaces whose forwarding state is described with flags. MRIB communication is based on the state change of entry and interface flags. Flags are significant to MRIB clients and not interpreted by MRIB.

23.2.4 Hardware Dependent Forwarding and Fast Dropping

In IP multicast protocols, each (S,G) and (*,G) route corresponds to an inbound reverse path forwarding (RPF) interface. Packets arriving on non-RPF interfaces may require PIM processing, as performed by the CPU subsystem software.

By default, hardware sends all packets arriving on non-RPF interfaces to the CPU subsystem software. However, the CPU can be overwhelmed by non-RPF packets that do not require software processing. The CPU subsystem software prevents CPU overload by creating a fast-drop entry in hardware for inbound non-RPF packets not requiring PIM processing. Packets matching a fast-drop entry is bridged in the ingress VLAN, but not sent to the software, avoiding CPU subsystem software overload. Fast-drop entry usage is critical in topologies with persistent RPF failures.

Protocol events, such as links going down or unicast routing table changes, can change the set of packets that can be fast dropped. Packets that were correctly fast dropped before a topology change may require forwarding to the CPU subsystem software after the change. The CPU subsystem software handles fast-drop entries that respond to protocol events so that PIM can process all necessary non-RPF packets.

23.3 Multicast Protocols

23.3.1 IGMP

Networks use Internet Group Management Protocol (IGMP) to control the flow of layer 3 multicast traffic. Hosts request and maintain multicast group membership through IGMP messages. Multicast routers use IGMP to maintain a membership list of active multicast groups for each attached network.

- IGMP version 1 is defined in RFC 1112. Hosts could join multicast groups but had no mechanism to signal a request to leave a group. Routers use a time-out based process to determine the groups to which the hosts had lost interest.
- IGMP version 2 is defined in RFC 2236. Version 2 added leave messages that hosts use to terminate group membership.
- IGMP version 3 is defined in RFC 4604. Version 3 allows hosts to specify IP addresses within a group from which it receives traffic. Traffic from all other group addresses is block from the host.

With respect to each of its attached networks, a multicast router is either a querier or non-querier. Each physical network contains only one querier. A network with more than one multicast router designates the router with the lowest IP address as its querier.

Queriers solicit group membership information by periodically sending General Query messages. Queriers also receive unsolicited messages from hosts joining or leaving a multicast group. When a querier receives a message from a host, it updates its membership list for the group referenced in the message and the network where the message originated.

Queriers forward multicasts from remote sources only to networks as specified by its membership list. If a querier does not receive a report from a network host for a specific group, it removes the corresponding entry from the table and discontinues forwarding multicasts for that group on the network. Queriers also send group-specific queries after receiving a leave request from a host to determine if the network still contains active multicast group members. If it does not receive a membership report during the period defined by the ***last member query response interval***, the querier removes the group-network entry from the membership list.

When a host receives a General Query, it responds with Membership Report messages for each of its multicast groups within the interval specified by the Max Response Time field in the query. IGMP suppresses multiple messages from different hosts on a network for the same group. Hosts send unsolicited Membership reports to join a multicast group and send leave messages to exit a group.

23.3.2 IGMP Snooping

IGMP snooping is a layer 2 optimization for the layer 3 IGMP protocol. IGMP snooping takes place internally on switches and is not a protocol feature. IGMP snooping prevents local network hosts from receiving traffic for multicast groups they did not join and prunes multicast traffic from links that do not contain IGMP clients.

When snooping is enabled, a switch analyzes IGMP packets between hosts connected to network switches and multicast routers (mrouters). When a switch finds an IGMP Report from a multicast group recipient, it adds the recipient's port to the group multicast list. When the switch receives an IGMP leave, it removes the recipient's port from the list. Groups are removed upon the group timer expiry.

Snooping requires an IGMP querier in the network. Tables created for snooping are associated with the querier. Without a querier the tables are not created and snooping does not work. An IGMP snooping querier performs the multicast router (mrouter) role when the network does not have a router. When the querier is enabled on a VLAN, the switch periodically broadcasts IGMP queries and listens for IGMP Reports that indicate host group memberships.

A static mrouter can be configured for a specific port. Static mrouters are not learned through snooping. Any data port can act as a static mrouter. When a static mrouter is configured, it replaces any dynamic mrouters learned through IGMP snooping.

When a network contains multiple mrouters, they elect one as the querier, based on IP address. When IGMP querier is enabled on a VLAN, the switch performs as a querier only if it is elected or it is the only querier on the network.

23.3.3 PIM-SM

Protocol Independent Multicast (PIM) is a collection of multicast routing protocols, each optimized for a different environment. PIM Sparse Mode (PIM-SM), defined in RFC 4601, is a multicast routing protocol designed for networks where multicast group recipients are sparsely distributed, including wide-area and inter-domain networks.

PIM builds and maintains multicast routing trees using reverse path forwarding (RPF) on a unicast routing table. PIM can use routing tables consisting of EIGRP, OSPF, BGP, and static routes. All sources send traffic to the multicast group through shared trees that have a common root node called the Rendezvous Point (RP). Each host (senders and receivers) is associated with a Designated Router (DR) that acts for all directly connected hosts in PIM-SM transactions.

23.3.3.1 Protocol Overview

PIM uses an MRIB that is populated from the unicast table. The MRIB provides the next-hop router along a multicast-capable path to each destination subnet. This determines the next-hop neighbor for sending PIM Join or Prune messages.

PIM establishes multicast routes through three phases:

- Establishing the RP Tree
- Eliminating Encapsulation
- Establishing the Shortest Path Tree (SPT)

Establishing the RP Tree (Phase 1)

The RP tree is a distribution network that all sources share to deliver multicast data. The root of the RP tree is the Rendezvous Point.

The process starts when a receiver requests multicast data from a group (G). The receiver's DR sends a PIM (*,G) Join message toward the multicast group's RP. As the message travels towards the RP, it instantiates the multicast (*,G) state in each router on the path. After many receivers join the group, the Join messages converge on the RP to form the RP tree.

The DR resends Join messages periodically, while it has a receiver in the group, to prevent state timeout expiry in the routers along the path. When all receivers on a DR's subnet leave a group, the DR sends a (*,G) Prune message towards the RP to remove the state from the routers.

A multicast sender transmits multicast data to the RP through its DR. The DR encapsulates the multicast packets and sends them as unicast packets. The RP extracts the native (unencapsulated) multicast packet and sends it to the RP tree towards the group members.

Eliminating Multicast Encapsulation (Phase 2)

Data encapsulation, while initially required before the multicast path is established, is inefficient because it requires the transmission of data that is extraneous to multicast. Phase 2 establishes states in the routers that support the transmission of native multicast packets.

When the RP receives an encapsulated packet from source S on group G, it sends a source-specific (S,G) join message towards the source. As the message travels towards S, it instantiates the (S,G) state on each router in the path. This state is used only to forward packets for group G from source S. Data packets on the (S,G) path are also routed into the RP tree when they encounter an (*,G) router.

When the RP starts receiving native packets from the sources, it sends a Register-Stop message to the source's DR, halting packet encapsulation. At this time, traffic flows natively from the source along a source-specific tree to the RP, then along the shared RP tree to the receivers.

Establishing the Shortest Path Tree (Phase 3)

The third phase establishes the shortest path from the multicast source to all receivers.

When a multicast packet arrives at the receiver, its router (typically the DR) sends a Join message towards the source to instantiate the (S,G) state in all routers along its path. The message eventually reaches either the source's subnet or a router that already has an (S,G) state. This causes data to flow from the source to the receiver following the (S,G) path. At this time, the receiver is receiving data from the Shortest Path Tree (SPT) and the RPT.

The DR (or upstream router) eliminates the data transmission along the RPT by sending a Prune message (S,G,rpt) towards the RP. The message travels hop-by-hop, instantiating the state on each router in the path, continues until it reaches the RP or a router that needs traffic from S for other receivers.

23.3.3.2 Rendezvous Points (RP)

An RP is a router that is configured as the root of a non-source-specific distribution tree for a multicast group. Join messages for receivers to a group are sent towards the RP. Data from senders is sent to the RP, allowing receivers to discover sender identity and begin receiving group traffic. Paths through RP routers are temporary; when traffic volume reaches a sufficient level, the receiver joins a source-specific tree and the path through the RP is dropped.

The switch supports two method of mapping RPs to multicast groups:

- Static: RPs are statically configured through a CLI statement.
- Dynamic: RPs are dynamically selected by a bootstrap router from a set of candidate RPs .

While dynamic RP mappings have priority over static maps by default, a static RP can be configured to override dynamic mappings.

PIM Anycast-RP defines a single RP address that is configured on multiple routers. An anycast-RP set consists of the routers configured with the same anycast-RP address. Anycast-RP provides redundancy protection and load balancing. The anycast-RP set supports all multicast groups.

Section 23.4.4.1 describes the configuration of rendezvous points.

23.4 Configuring Multicast

This section describes the following configuration tasks:

- Section 23.4.1: Enabling Multicast Routing, PIM, IGMP, and IGMP Snooping
- Section 23.4.2: Configuring IGMP
- Section 23.4.3: Configuring IGMP Snooping
- Section 23.4.4: Configuring PIM Parameters

23.4.1 Enabling Multicast Routing, PIM, IGMP, and IGMP Snooping

23.4.1.1 Enabling Multicast Routing

Enabling IP multicast routing allow the switch to forward multicast packets. The **ip multicast-routing** command enables multicast routing. When multicast routing is enabled, ***running-config*** contains an **ip multicast-routing** statement.

Example

- This command enables multicast routing on the switch.

```
Switch(config)#ip multicast-routing
Switch(config)#
```

23.4.1.2 Enabling PIM and IGMP

Enabling PIM on an interface also enables IGMP on that interface. When the switch populates the multicast routing table, interfaces are added to the table only when periodic join messages are received from downstream routers, or when there is a directly connected member on the interface.

When forwarding from a LAN, sparse-mode operates if a rendezvous point is known for the group. Packets are encapsulated and sent toward the RP. When no RP is known, the packet is flooded. If the multicast traffic from a specific source is sufficient, the receiver's first-hop router can send join messages toward the source to build a source-based distribution tree.

By default, PIM is disabled on an interface. The **ip pim sparse-mode** command enables PIM on the active interface.

Example

- This command enables PIM and IGMP on VLAN interface 8.

```
Switch(config-if-Vl8)#ip pim sparse-mode
Switch(config-if-Vl8)#
```

23.4.1.3 Enabling Snooping

The switch provides two control settings for snooping IGMP packets:

- VLAN settings manage snooping on individual VLANs.
  - When global snooping is enabled, snooping can be enabled or disabled on individual VLANs.
  - When global snooping is disabled, snooping cannot be enabled on individual VLANs.
- Global settings control snooping on the interfaces where VLAN settings are not configured.
  - Snooping is globally enabled by default.

The **ip igmp snooping** command controls the global snooping setting. The **ip igmp snooping vlan** command enables snooping on individual VLANs if snooping is globally enabled. IGMP snooping is enabled on all VLANs by default.

Example

- This command globally enables snooping on the switch.

```
switch(config)#ip igmp snooping
```

- This command disables snooping on VLANs 2 through 4.

```
switch(config)#no ip igmp snooping vlan 2-4
```

23.4.2 Configuring IGMP

23.4.2.1 Configuring IGMP Settings

An interface that runs IGMP uses default protocol settings unless otherwise configured. The switch provides commands that alter startup query, last member query, and normal query settings.

IGMP Version

The switch supports IGMP versions 1 through 3. The **ip igmp version** command configures the IGMP version on the configuration mode interface. Version 3 is the default IGMP version.

Example

- This command configures IGMP version 3 on VLAN interface 4

```
switch(config-if-Vl4)#ip igmp version 3
switch(config-if-Vl4)#
```

Startup Query

Membership queries are sent at an increased frequency immediately after an interface starts up to quickly establish the group state. Query count and Query interval commands adjust the period between membership queries for a specified number of messages.

The **ip igmp startup-query-interval** command specifies the interval between membership queries that an interface sends immediately after it starts up. The **ip igmp startup-query-count** command specifies the number of queries that the switches sends from the interface at the startup interval rate.

Example

- These commands define a startup interval of 15 seconds for the first 10 membership queries sent from VLAN interface 12.

```
Switch(config-if-Vl12)#ip igmp startup-query-interval 150
Switch(config-if-Vl12)#ip igmp startup-query-count 10
Switch(config-if-Vl12)#
```

Membership Queries

The router with the lowest IP address on a subnet sends membership queries as the IGMP querier. When a router receives a membership query from a source with a lower IP address, it resets its query response timer. Upon timer expiry, the router begins sending membership queries. If the router subsequently receives a membership query from a router with a lower IP address, it stops sending membership queries and resets the query response timer.

The **ip igmp query-interval** command configures the frequency at which the active interface, as an IGMP querier, sends membership query messages.

The **ip igmp query-max-response-time** command configures the time that a host has to respond to a membership query.

Example

- These commands define a Membership query interval of 75 seconds and a query response timer reset value of 45 seconds for queries sent from VLAN interface 15.

```
Switch(config-if-Vl15)#ip igmp query-interval 75
Switch(config-if-Vl15)#ip igmp query-max-response-time 450
Switch(config-if-Vl15)#
```

Last Member Query

When the querier receives an IGMP leave message, it verifies the group has no remaining hosts by sending a set of group-specific queries at a specified interval. If the querier does not receive a response to the queries, it removes the group state and discontinues multicast transmissions.

The **ip igmp last-member-query-count** (LMQC) command specifies the number of query messages the router sends in response to a group-specific or group-source-specific leave message.

The **ip igmp last-member-query-interval** command configures the transmission interval for sending group-specific or group-source-specific query messages to the active interface.

Example

- These commands program the switch to send 3 query messages, one every 25 seconds, when VLAN interface 15 receives an IGMP leave message.

```
Switch(config-if-Vl15)#ip igmp last-member-query-interval 250
Switch(config-if-Vl15)#ip igmp last-member-query-count 3
Switch(config-if-Vl15)#
```

Static Groups

The **ip igmp static-group** command configures the active interface as a static member of the specified multicast group. The router forwards multicast group packets through the interface without otherwise appearing or acting as a group member. By default, no static group membership entries are configured on interfaces.

Example

- This command configures VLAN interface 5 as a static member of the multicast group at address 241.1.1.15 for multicast data packets that originate at 15.1.1.1.

```
switch(config-if-Vl5)#ip igmp static-group 241.1.1.45 15.1.1.1
```

23.4.3 Configuring IGMP Snooping

IGMP snooping is an IP multicast constraining mechanism that runs on a Layer 2 switch. The switch examines join/leave messages from IGMP packets sent between the hosts and the router. When the switch finds an IGMP report from a host for a multicast group, it adds the port number of the host to the associated multicast table entry. When the switch finds an IGMP leave group message from a host, it removes the table entry of the host. The switch uses this table to direct multicast packets to only hosts that are members of the packet's destination group.

23.4.3.1 IGMP Snooping Filters

IGMP Snooping filters controls the multicast groups that an interface can join through IGMP profiles. An IGMP profile, which is applied to Ethernet and port channel interfaces, specifies a filter type and a list of address ranges. The address ranges comprise the multicast groups covered by the profile. The filter type determines an interface's accessiblity to the multicast groups:

- Permit filters define the multicast group the interface can join.

- Deny filters define the multicast groups the interface cannot join.

Profiles are created in IGMP-profile configuration mode, then applied to an interface from the configuration mode for that interface.

The **ip igmp profile** command places the switch in IGMP profile configuration mode. The **permit / deny** and **range** commands specify the profile's filter type and address range. A profile may contain multiple range statements to define a discontiguous address range.

Example

- These commands creates an IGMP profile name list_1 by enter IGMP-profile configuration mode, then configures the profile to permit multicast groups 231.22.24.0 through 231.22.24.127. The switch is returned to global configuration mode after the profile is created.

```
Switch(config)#ip igmp profile list_1
Switch(config-igmp-profile-list_1)#permit
Switch(config-igmp-profile-list_1)#range 231.22.24.0 231.22.24.127
Switch(config-igmp-profile-list_1)#exit
Switch(config)#
```

The **ip igmp snooping filter** command applies an IGMP snooping profile to a configuration mode interface.

- These commands applies the ***list_1*** snooping profile to Ethernet interface 7.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#ip igmp snooping filter list_1
switch(config-if-Et7)#
```

23.4.3.2 Enabling the Snooping Querier

The IGMP snooping querier supports snooping by sending layer 2 membership queries to hosts attached to the switch. QoS does not support IGMP packets when IGMP snooping is enabled.

Enabling the snooping querier on an interface requires the explicit configuration of a global querier address or a local querier address for the interface. See Section 23.4.3.3: Configuring the Snooping Querier.

The switch provides two control settings for controlling the snooping querier:

- The global setting controls the querier on VLANs for which there is no snooping querier command.
- VLAN querier settings take precedence over the global querier setting.

The **ip igmp snooping querier** command controls the global querier setting. When enabled globally, the querier is controlled on individual VLANs through the **ip igmp snooping vlan querier** command.

The **ip igmp snooping vlan querier** command controls the querier for the specified VLANs. VLANs follow the global querier setting unless overridden by one of these commands:

- **ip igmp snooping vlan querier** enables the querier on specified VLANs.
- **no ip igmp snooping vlan querier** disables the querier on specified VLANs.

Example

- These commands globally enables the snooping querier on the switch, explicitly disables snooping on VLANs 1-4, and explicitly enables snooping on VLANs 5-8.

```
switch(config)#ip igmp snooping querier
switch(config)#no ip igmp snooping vlan 1-4 querier
switch(config)#ip igmp snooping vlan 5-8 querier
```

- This command removes the querier setting for VLANs 3-6:

```
switch(config)#default ip igmp snooping vlan 3-6 querier
```

23.4.3.3 Configuring the Snooping Querier

Querier Address

The switch provides two IP addresses for setting the querier source:

- The global address is used by VLANs for which there is no querier address command.
- VLAN querier address settings take precedence over the global querier address.

The snooping querier address specifies the source IP address for IGMP snooping query packets transmitted by the switch. The source address is also used to elect a snooping querier when the subnet contains multiple snooping queriers.

The default global querier address is not defined. When the configuration includes a snooping querier, a querier address must be defined globally or for each interface that enables a querier.

The **ip igmp snooping querier address** command sets the global querier source IP address for the switch. VLANs use the global address unless overwritten with the **ip igmp snooping vlan querier address** command. The default global address is not defined.

The **ip igmp snooping vlan querier address** command sets the source IP address for query packets transmitted from the specified VLAN. This command overrides the **ip igmp snooping querier address** for the specified VLAN.

Examples

- This command sets the source IP address for query packets transmitted from the switch to 10.1.1.41

```
switch(config)#ip igmp snooping querier address 10.1.1.41
```

- This command sets the source IP address for query packets transmitted from VLAN 2 to 10.14.1.1.

```
switch(config)#ip igmp snooping vlan 2 querier address 10.14.1.1
```

Membership Query Interval

The query interval is the period (seconds), between IGMP Membership Query message transmissions. The default query interval is 125 seconds.

The **ip igmp snooping querier query-interval** command specifies the global query-interval for packets sent from a snooper querier. Values range from 5 to 3600 seconds. The default global setting is 125 seconds. VLANs use the global setting unless overwritten with the **ip igmp snooping vlan querier query-interval** command.

The **ip igmp snooping vlan querier query-interval** command specifies the query interval for packets sent from the snooping querier to the specified interface, overriding the global setting.

Examples

- This command sets a query interval of 150 seconds for queries transmitted from VLANs for which a query interval is not configured.

```
switch(config)#ip igmp snooping querier query-interval 150
```

- This command sets the query interval of 240 seconds for queries transmitted from VLAN 2.

```
switch(config)#ip igmp snooping vlan 2 querier query-interval 240
```

Membership Query Response Interval

The Max Response Time field, in Membership Query messages, specifies the longest time a host can wait before responding with a Membership Report message. In all other messages, the sender sets the field to zero and the receiver ignores it. The switch provides two values for setting this field:

- The global value is used by VLANs for which there is no Max Response Time command.
- VLAN values take precedence over the global value for the specified VLAN.

The **ip igmp snooping querier max-response-time** command specifies the global Max Response Time value used in snooping query packets transmitted from the switch. Values range from 1 to 25 seconds with a default of 10 seconds. VLANs use the global setting unless overwritten with the **ip igmp snooping vlan querier max-response-time** command.

The **ip igmp snooping vlan querier max-response-time** command specifies the Max Response Time field contents for packets transmitted to the specified VLAN, overriding the global setting.

Examples

- This command sets the maximum response time of 15 seconds for queries transmitted from VLANs for which a maximum response time is not configured.

```
switch(config)#ip igmp snooping querier max-response-time 15
```

- This command sets a maximum response time of 5 seconds for queries transmitted from VLAN 2.

```
switch(config)#ip igmp snooping vlan 2 querier max-response-time 5
```

Robustness Variable

The robustness variable specifies the number of unacknowledged snooping queries that a switch sends before removing the recipient from the group list.

The **ip igmp snooping robustness-variable** command configures the robustness variable for all snooping packets sent from the switch. The default value is 2.

Example

- This command sets the robustness-variable value to 3.

```
switch(config)#ip igmp snooping robustness-variable 3
```

Configuring the Network

The **ip igmp snooping vlan mrouter** command statically configures a port that connects to a multicast router to join all multicast groups. The port to the router must be in the specified VLAN range.

Snooping may not always be able to locate the IGMP querier. This command is for IGMP queriers that are known to connect through the network to a port on the switch.

Example

- This command configures the static connection to a multicast router through Ethernet port 3.

```
switch(config)#ip igmp snooping vlan 2 mrouter interface ethernet 3
```

The **ip igmp snooping vlan static** command adds an a port to a multicast group. The IP address must be an unreserved IPv4 multicast address. The interface to the port must be in the specified VLAN range.

Example

- This command configures the static connection to a multicast group at 224.2.1.4 through Ethernet port 3.

```
switch(config)#ip igmp snooping vlan 2 static 224.2.1.4 interface ethernet 3
```

23.4.4 Configuring PIM Parameters

The following sections describe the configuration of static RPs, dynamic RPs, and anycast-RPs. RP implementation is defined through the following RFCs:

- RFC 5059: Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)
- RFC 6226: PIM Group-to-Rendezvous-Point Mapping

23.4.4.1 Rendezvous Points (RPs)

Networks that run PIM sparse mode require a rendezvous point (RP). The switch supports dynamic RPs, static RPs, and anycast-RP.

Configuring Static RPs

The **ip pim rp-address** command configures a static RP, providing an option to override dynamic RPs.

Examples

- This command creates a static RP at 172.17.255.83 that maps to all multicast groups (224/4) and override dynamic RPs.

```
Switch(config)#ip pim rp-address 172.17.255.83 override
Switch(config)#
```

- This command creates a static RP at 169.21.18.23 that maps to the multicast groups at 238.1.12.0/24.

```
Switch(config)#ip pim rp-address 169.21.18.23 238.1.12.0/24
Switch(config)#
```

Configuring Dynamic RPs

Dynamic RP selection is implemented through a Bootstrap Router (BSR), which is a PIM router within the PIM domain that selects RPs from a list of candidates. A subset of PIM routers within the domain are configured as candidate Bootstrap routers (C-BSRs). Through the exchange of Bootstrap messages (BSMs), the C-BSRs elect the BSR, which then uses BSMs to inform all domain routers of its status.

The BSR holdtime defines the timeout period that an elected BSR remains valid after the receipt of a BSM and is also used in dynamic RP configuration. Holdtime is designated by the BSR router and communicated to other routers through BSMs.

Another subset of domain PIM routers are configured as candidate RPs (C-RPs). The BSR creates a set of qualifying RPs from the list of C-RPs, then distributes the group-to RP mapping set to all domain routers through BSMs. Each PIM router, after receiving this set, uses a standard algorithm defined in RFC 6226 to select one RP per multicast group.

The **ip pim bsr-candidate** command configures the switch as a candidate BSR router (C-BSR). Command parameters specify the switch's BSR address, the interval between BSM transmissions, and the switch's BSR priority rating. Priority ratings range from 0 to 255 with a default of 128. Higher numbers denote higher priority during BSR elections.

Example

- These commands configure the switch as a BSR candidate, using the IP address assigned to VLAN interface 24 as its BSR address. The BSM transmission interval is set to 30 seconds and the priority is set to 192.

```
switch(config)#ip pim bsr-candidate vlan 24 priority 192 interval 30
switch(config)#
```

The **ip pim bsr-holdtime** command specifies the value the switch inserts in the ***holdtime*** field of Bootstrap messages (BSMs) that it sends. This value becomes the holdtime for the PIM domain if the switch is elected as the BSR.

Example

- This command specifies 75 seconds as the value that the switch inserts into BSM holdtime fields.

```
switch(config)#ip pim bsr-holdtime 75
switch(config)#
```

The **ip pim rp-candidate** command configures the switch as a candidate rendezvous point (C-RP) . The BSR selects a multicast group's dynamic RP set from the list of C-RPs. Command parameters specify the switch's RP address, C-RP advertisement interval, and priority rating. The priority rating is used by the BSR when selecting RPs. The C-RP advertisement interval specifies the period between successive C-RP advertisement message transmissions to the BSR.

Running-config may contain multiple **ip pim rp-candidate** statements to support multiple multicast groups:

- All commands must specify the same interface. Issuing a command with an interface that differs from existing commands removes all existing commands from ***running-config***.
- ***Running-config*** stores the ***interval*** setting in a separate statement that applies to all **rp-candidate** statements. Commands that specify an interval that differs from the previously configured value place the new value in ***running-config***. This new value applies to all **rp-candidate** statements.

Example

- This command configures a switch as a candidate RP for the multicast group 235.1.1.0/24, with a priority of 48 and a RP advertisement interval of 45 seconds.

```
Switch(config)#ip pim rp-candidate vlan 24 235.1.1.0/24 priority 48
interval 45
Switch(config)#
```

By default, the switch transmits bootstrap router messages (BSMs) over all PIM-SM enabled interfaces. The **ip pim bsr-border** command prevents the switch from transmitting BSMs over the configuration mode interface.

Example

- This command creates a static RP at 172.17.255.83 that maps to all multicast groups (224/4) and override dynamic RPs.

```
Switch(config)#interface vlan 10
switch(config-if-Vl10)#ip pim bsr-border#
switch(config-if-Vl10)
```

Anycast-RP

PIM Anycast-RP defines a single RP address that is configured on multiple routers. An anycast-RP set consists of the routers configured with the same anycast-RP address. Anycast-RP provides redundancy protection and load balancing. The anycast-RP set supports all multicast groups.

PIM register messages are unicast to the RP by designated routers (DRs) that are directly connected to multicast sources. The switch sends these messages and join-prune messages to the anycast-RP set member specified in the anycast-RP command. In a typical configuration, one command is required for each member of the anycast-RP set.

The PIM register message has the following functions:

- Notify the RP that a source is actively sending to a multicast group.

- Deliver multicast packets sent by the source to the RP for delivery down the shared tree.

The DR continues sending PIM register messages to the RP until it receives a Register-Stop message from the RP. The RP sends a Register-Stop message in either of the following cases:

- The RP has no receivers for the multicast group being transmitted.
- The RP has joined the SPT to the source but has not started receiving traffic from the source.

The **ip pim anycast-rp** command configures the switch as a member of an anycast-RP set and establishes a communication link with another member of the set.

Example

- These commands configure a switch (IP address 10.1.1.14) into an anycast-RP set with an RP address of 172.17.255.29. The anycast-RP set contains three other routers, located at 10.1.2.14, 10.1.3.14, and 10.1.4.14. It sets the number of unacknowledged register messages it sends to each router at 15.

```
Switch(config)#ip pim anycast-rp 172.17.255.29 10.1.1.14 register-count 15
Switch(config)#ip pim anycast-rp 172.17.255.29 10.1.2.14 register-count 15
Switch(config)#ip pim anycast-rp 172.17.255.29 10.1.3.14 register-count 15
Switch(config)#ip pim anycast-rp 172.17.255.29 10.1.4.14 register-count 15
```

23.4.4.2 Hello Messages

Multicast routers send PIM router query (Hello) messages to determine the designated router (DR) for each subnet. The DR sends Internet Group Management Protocol (IGMP) host query messages to all hosts on the directly connected LAN and source registration messages to the RP.

The **ip pim query-interval** command specifies the transmission interval between PIM hello messages originating from the specified VLAN interface.

Example

- This command configures 45 second intervals between hello messages originating from VLAN interface 4.

```
Switch(config-if-Vl4)#ip pim query-interval 45
Switch(config-if-Vl4)#
```

23.4.4.3 Designated Router Election

PIM uses these criteria for electing designated routers (DR):

- If one router does not advertise a dr-priority value, the router with the highest IP address becomes the Designated Router.
- If all routers advertise a dr-priority value, the router with the highest dr-priority value becomes the Designated Router.

The **ip pim dr-priority** command sets the DR priority value that the switch advertises. If running-config does not contain a **ip pim dr-priority** statement, the switch does not advertise a dr-priority value.

Examples

- This command configures the dr-priority value of 15 on VLAN interface 4.

```
Switch(config-if-Vl4)#ip pim dr-priority 15
Switch(config-if-Vl4)#
```

- This command removes the **ip-pim dr-priority** statement (VLAN interface 4) from running-config.

```
Switch(config-if-Vl4)#no ip pim dr-priority
Switch(config-if-Vl4)#
```

23.4.4.4 Join-Prune Messages

A Designated Router (DR) sends periodic Join/Prune messages toward a group-specific Rendezvous Point (RP) for each group for which it has active members. These messages inform other PIM routers about clients that want to become receivers (Join) or stop being receivers (Prune) for the group groups.

The **ip pim join-prune-interval** command specifies the period between join/prune messages that the switch originates from the specified VLAN interface and sends to the upstream RPF neighbor.

Example

- This command configures 75 second intervals between join/prune messages originating from VLAN interface 4.

```
Switch(config-if-Vl4)#ip pim join-prune-interval 75
Switch(config-if-Vl4)#
```

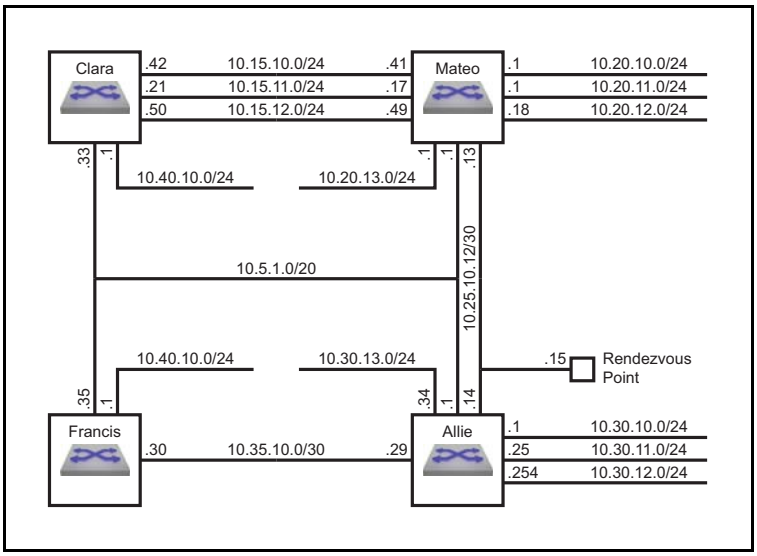
23.5 Multicast Example

This section provides an example network that implements multicast and includes the required commands.

23.5.1 Diagram

Figure 23-2 displays the multicast network example. The network contains four routers. Multicast routing is enabled on two switches. One switch has its querier enabled.

Figure 23-2 Multicast Example



The example multicast network implements these multicast parameters:

Rendezvous Point Address: 10.25.10.15

Switch Clara

- Snooping: disabled
- Subnet Summary:
  - 10.40.10.0/24: VLAN 11
  - 10.15.10.0/24: VLAN 12
  - 10.15.11.0/24: VLAN 13
  - 10.15.12.0/24: VLAN 14
  - 10.5.1.0/20: VLAN 10

Switch Mateo

- Snooping: disabled
- Subnet Summary:
  - 10.20.13.0/24: VLAN 18
  - 10.20.10.0/24: VLAN 15
  - 10.20.11.0/24: VLAN 16
  - 10.20.12.0/24: VLAN 17
  - 10.15.10.0/24: VLAN 12

— 10.15.11.0/24: VLAN 13
— 10.15.12.0/24: VLAN 14
— 10.25.10.12/30: VLAN 19
— 10.5.1.0/20: VLAN 10

Switch Allie

- Snooping: enabled
- Multicast Routing: enabled
- Querier: enabled
- Rendezvous Point Address: 10.25.10.15
- MFIB activity polling interval: 5 second
- Subnet Summary:
  — 10.30.13.0/24: VLAN 23
  — 10.30.10.0/24: VLAN 20 – PIM-SM enabled
  — 10.30.11.0/24: VLAN 21 – PIM-SM enabled
  — 10.30.12.0/24: VLAN 22
  — 10.25.10.12/30: VLAN 19
  — 10.35.10.0/30: VLAN 24 – PIM-SM enabled
  — 10.5.1.0/20: VLAN 10

Switch Francis

- Snooping: enabled
- Multicast Routing: enabled
- Subnet Summary:
  — 10.40.10.0/24: VLAN 25 – PIM-SM enabled
  — 10.35.10.0/30: VLAN 24 – PIM-SM enabled
  — 10.5.1.0/20: VLAN 10

23.5.2 Code

This code configures multicasting.

Step 1 Configure the interface addresses

Step a Router Clara interfaces

```
Clara(config)#interface vlan 11
Clara(config-if-vl11)#ip address 10.40.10.1/24
Clara(config-if-vl11)#interface vlan 12
Clara(config-if-vl12)#ip address 10.15.10.42/24
Clara(config-if-vl12)#interface vlan 13
Clara(config-if-vl13)#ip address 10.15.11.21/24
Clara(config-if-vl13)#interface vlan 14
Clara(config-if-vl14)#ip address 10.15.12.50/24
Clara(config-if-vl14)#interface vlan 10
Clara(config-if-vl10)#ip address 10.5.1.33/20
Clara(config-if-vl10)#router ospf 1
Clara(config-router-ospf)#redistribute static
```

Step b Router Mateo interfaces

```
Mateo(config)#interface vlan 18
Mateo(config-if-vl18)#ip address 10.20.13.1/24
Mateo(config-if-vl18)#interface vlan 15
Mateo(config-if-vl15)#ip address 10.20.10.1/24
Mateo(config-if-vl15)#interface vlan 16
Mateo(config-if-vl16)#ip address 10.20.11.1/24
Mateo(config-if-vl16)#interface vlan 17
Mateo(config-if-vl17)#ip address 10.20.12.16/24
Mateo(config-if-vl17)#interface vlan 12
Mateo(config-if-vl12)#ip address 10.15.10.41/24
Mateo(config-if-vl12)#interface vlan 13
Mateo(config-if-vl13)#ip address 10.15.11.17/24
Mateo(config-if-vl13)#interface vlan 14
Mateo(config-if-vl14)#ip address 10.15.12.49/24
Mateo(config-if-vl14)#interface vlan 19
Mateo(config-if-vl19)#ip address 10.25.10.13/30
Mateo(config-if-vl19)#interface vlan 10
Mateo(config-if-vl10)#ip address 10.5.1.1/20
Mateo(config-if-vl10)#router ospf 1
Mateo(config-router-ospf)#redistribute static
```

Step c Router Allie interfaces

```
Allie(config)#interface vlan 23
Allie(config-if-vl23)#ip address 10.30.13.34/24
Allie(config-if-vl23)#interface vlan 20
Allie(config-if-vl20)#ip address 10.30.10.1/24
Allie(config-if-vl20)#interface vlan 21
Allie(config-if-vl21)#ip address 10.30.11.25/24
Allie(config-if-vl21)#interface vlan 22
Allie(config-if-vl22)#ip address 10.30.12.254/24
Allie(config-if-vl22)#interface vlan 19
Allie(config-if-vl19)#ip address 10.25.10.14/30
Allie(config-if-vl19)#interface vlan 24
Allie(config-if-vl24)#ip address 10.35.10.29/30
Allie(config-if-vl24)#interface vlan 10
Allie(config-if-vl10)#ip address 10.5.1.1/20
Allie(config-if-vl10)#router ospf 1
Allie(config-router-ospf)#redistribute static
```

Step d Router Francis interfaces

```
Francis(config)#interface vlan 25
Francis(config-if-vl25)#ip address 10.40.10.1/24
Francis(config-if-vl25)#interface vlan 24
Francis(config-if-vl24)#ip address 10.35.10.30/24
Francis(config-if-vl24)#interface vlan 10
Francis(config-if-vl10)#ip address 10.5.1.35/24
Francis(config-if-vl10)#router ospf 1
Francis(config-router-ospf)#redistribute static
```

Step 2 Configure the interface multicast parameters

Step a Router Allie interfaces

```
Allie(config-router-ospf)#interface vlan 20
Allie(config-if-vl20)#ip pim sparse-mode
Allie(config-if-vl20)#interface vlan 21
Allie(config-if-vl21)#ip pim sparse-mode
Allie(config-if-vl21)#interface vlan 24
Allie(config-if-vl24)#ip pim sparse-mode
Allie(config-if-vl24)#interface vlan 10
Allie(config-if-vl10)#ip pim sparse-mode
```

Step b Router Francis interfaces

```
Francis(config-router-ospf)#interface vlan 25
Francis(config-if-vl25)#ip pim sparse-mode
Francis(config-if-vl25)#interface vlan 24
Francis(config-if-vl24)#ip pim sparse-mode
```

Step 3 Configure the router multicast parameters

Step a Router Clara parameters

```
Clara(config-router-ospf)#exit
Clara(config)#no ip igmp snooping
```

Step b Router Mateo interfaces

```
Mateo(config-router-ospf)#exit
Mateo(config)#no ip igmp snooping
```

Step c Router Allie interfaces

```
Allie(config-if-vl10)#exit
Allie(config)#ip multicast-routing
Allie(config)#ip mfib activity polling-interval 5
Allie(config)#ip pim rp-address 10.25.10.15
```

Step d Router Francis interfaces

```
Francis(config-if-vl24)#exit
Francis(config)#ip multicast-routing
Francis(config)#ip pim rp-address 10.25.10.15
```

23.6 Multicast Commands

This section contains descriptions of the CLI commands that this chapter references.

Multicast Configuration Commands (Global)

- ip mfib activity polling-interval . . . . . Page 1015
- ip mfib max-fastdrops . . . . . Page 1017
- ip multicast-routing . . . . . Page 1019

Multicast Configuration Commands (Interface)

- ip mfib fastdrop . . . . . Page 1016
- ip multicast boundary . . . . . Page 1018

Multicast Clear Commands

- clear ip mfib fastdrop . . . . . Page 1013
- clear ip mroute . . . . . Page 1014

Multicast Display Commands

To display the information in the multicast routing table, use the **show ip mroute** command. To display the MFIB table information, use the **show ip mfib** command.

- show ip mfib . . . . . Page 1020
- show ip mroute . . . . . Page 1022
- show ip mroute count . . . . . Page 1023

clear ip mfib fastdrop

The **clear ip mfib fastdrop** command removes all fast-drop entries from the MFIB table.

Command Mode

Privileged EXEC

Command Syntax

```
clear ip mfib fastdrop
```

Example

- This command removes all fast-drop entries from the MFIB table.

```
switch#clear ip mfib fastdrop
switch#
```

clear ip mroute

The **clear ip mroute** command removes route entries from the mroute table, as follows:

- **clear ip mroute *** all entries from the mroute table.
- **clear ip mroute** *gp_ipv4* all entries for the specified multicast group.
- **clear ip mroute** *gp_ipv4 src_ipv4* all entries for the specified source sending to a specified group.

Command Mode

Privileged EXEC

Command Syntax

```
clear ip mroute ENTRY_LIST
```

Parameters

- *ENTRY_LIST* entries that the command removes from the mroute table. Options include:
  - * all route entries are removed from the table
  - *gp_ipv4* all entries for multicast group *gp_ipv4* (dotted decimal notation).
  - *gp_ipv4 src_ipv4* all entries for source (*src_ipv4*) sending to group (*gp_ipv4*).

Examples

- This command removes all route entries from the mroute table.

```
switch#clear ip mroute *
switch#
```

- This command removes entries for the source 228.3.10.1 sending to multicast group 224.2.205.42.

```
switch#clear ip mroute 224.2.205.42 228.3.10.1
switch#
```

ip mfib activity polling-interval

The switch records activity levels for multicast routes in the mfib after polling the corresponding hardware activity bits. The **ip mfib activity polling-interval** command specifies the frequency that the switch polls the hardware activity bits for the multicast routes.

The **no ip mfib activity polling-interval** and **default ip mfib activity polling-interval** commands restore the default interval of 60 seconds by removing the **ip mfib activity polling-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip mfib activity polling-interval period
no ip mfib activity polling-interval
default ip mfib activity polling-interval
```

Parameters

- *period* interval (seconds) between polls. Values range from 1 to 60. Default is 60.

Example

- This command sets the MFIB activity polling period at 15 seconds.

```
switch(config)#ip mfib activity polling-interval 15
switch(config)#
```

ip mfib fastdrop

In IP multicast protocols, every (S,G) or (*,G) route is associated with an inbound RPF (reverse path forwarding) interface. Packets arriving on an interface not associated with the route may require specific PIM protocol processing performed by the CPU subsystem software. Therefore, all packets that arrive on a non-RPF interface are sent to the CPU subsystem software by default, which can overwhelm the CPU.

Multicast routing protocols often do not require non-RPF packets; these packets do not require software processing. The CPU subsystem software avoids unnecessary packet processing by loading fast-drop entries in the hardware when it receives an non-RPF interface packet that PIM does not require. Packets matching a fast-drop entry are bridged in the ingress VLAN, but not sent to the system software.

Fastdrop is enabled on all interfaces by default. The **no ip mfib fastdrop** command disables MFIB fast drops for the configuration mode interface.

The **ip mfib fastdrop** and **default ip mfib fastdrop** commands enable MFIB fast drops for the configuration mode interface by removing the corresponding **no ip mfib fastdrop** command from *running-config*.

The **clear ip mfib fastdrop** command, in global configuration mode, removes all MFIB fast drop entries on all interfaces.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip mfib fastdrop
no ip mfib fastdrop
default ip mfib fastdrop
```

Examples

- This command enables MFIB fast drops for the VLAN interface 120.

```
switch(config)#interface vlan 120
switch(config-if-Vl120)#ip mfib fastdrop
switch(config-if-Vl120)#
```

ip mfib max-fastdrops

The **ip mfib max-fastdrops** command limits the number of fast drop routes that the switch's MFIB table can contain.

The **no ip mfib max-fastdrops** and **default ip mfib max-fastdrops** commands restore the default fast drop route limit of 1024 by removing the **ip mfib max-fastdrops** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip mfib max-fastdrops quantity
no ip mfib mfib max-fastdrops
default ip mfib mfib max-fastdrops
```

Parameters

- *quantity* number of fast-drop routes. Value ranges from 0 to 1000000 (one million). Default is 1024.

Example

- This command sets the maximum number of fast drop routes at 2000.

```
switch(config)#ip mfib max-fastdrops 2000
switch(config)#
```

ip multicast boundary

The **ip multicast boundary** command specifies subnets where source traffic entering the configuration mode interface is filtered, preventing the creation of mroute states on the interface. The interface is not included in the outgoing interface list (OIL).

The multicast boundary can be specified through multiple IPv4 subnets or one standard IPv4 ACL.

Multicast pim, igmp or data packets are not allowed to flow across the boundary from either direction. The boundary facilitates the use of a multicast group address in different administrative domains.

The **no ip multicast boundary** and **default ip multicast boundary** commands delete the subnet restrictions by removing the **ip multicast boundary** command from the configuration

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip multicast boundary SUB_NET
no ip multicast boundary [SUB_NET]
default ip multicast boundary [SUB_NET]
```

Parameters

- *SUB_NET* the subnet address configured as the multicast boundary. Options include:
  - *net_addr* multicast subnet address (CIDR or address mask).
  - *acl_name* standard access control list (ACL) that specifies the multicast group addresses.

Restrictions

Multiple commands that specify a subnet can be assigned to an interface. No and default forms of the command must include the subnet being removed.

Only one command that specifies an ACL can be assigned to an interface. Commands that specify an ACL and a subnet cannot be simultaneously assigned to an interface.

Examples

- This command configures the multicast address of 229.43.23.0/24 as a multicast boundary where source traffic is restricted from VLAN interface 300.

```
switch(config)#interface vlan 300
switch(config-if-vl300)#ip multicast boundary 229.43.23.0/24
switch(config-if-vl300)#
```

- These commands create a standard ACL, then implements ACL in an ip multicast boundary command to configure two boundary subnets (225.123.0.0/16 and 239.120.10.0/24).

```
switch(config)#ip access-list standard mbac1
switch(config-std-acl-mbac1)#10 deny 225.123.0.0/16
switch(config-std-acl-mbac1)#20 deny 239.120.10.0/24
switch(config-std-acl-mbac1)#exit
switch(config)#interface vlan 200
switch(config-if-Vl200)#ip multicast boundary mbac1
switch(config-if-Vl200)#exit
switch(config)#
```

ip multicast-routing

The **ip multicast-routing** command allows the switch to forward multicast packets. Multicast routing is disabled by default.

The **no ip multicast-routing** and **default ip multicast-routing** commands disables multicast routing removing the **ip multicast-routing** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip multicast-routing
no ip multicast-routing
default ip multicast-routing
```

Example

- This command enables multicast routing on the switch.

```
switch(config)#ip multicast-routing
switch(config)#
```

show ip mfib

The **show ip mfib** command displays the forwarding entries and interfaces in the IPv4 Multicast Forwarding Information Base (MFIB):

- **show ip mfib** – displays MFIB information for hardware forwarded routes.
- **show ip mfib software** – displays MFIB information for software forwarded routes.

Command Mode

EXEC

Command Syntax

```
show ip mfib
show ip mfib software
```

Example

- This command displays MFIB information for hardware forwarded routes.

```
switch>show ip mfib
Activity poll time: 60 seconds
  239.255.255.250 172.17.26.25
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:02:11 ago
  239.255.255.250 172.17.26.156
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:02:11 ago
  239.255.255.250 172.17.26.178
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:03:37 ago
  239.255.255.250 172.17.26.190
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:02:11 ago
  239.255.255.250 172.17.26.209
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:02:11 ago
  239.255.255.250 172.17.26.223
    Vlan26 (iif)
    Vlan2028
    Cpu
      Activity 0:03:37 ago
switch>
```

- This command displays MFIB information for software forwarded routes.

```
switch>show ip mfib software
239.255.255.250 172.17.26.182
 pkts: 189 bytes: 55813 rpf failures: 0
    Vlan26 (iif)
    Pimreg
    Vlan2028
239.255.255.250 172.17.26.216
 pkts: 20 bytes: 3130 rpf failures: 0
    Vlan26 (iif)
    Pimreg
    Vlan2028
239.255.255.250 172.17.26.25
 pkts: 76 bytes: 12198 rpf failures: 0
    Vlan26 (iif)
    Pimreg
    Vlan2028
239.255.255.250 172.17.26.198
 pkts: 494 bytes: 77522 rpf failures: 0
    Vlan26 (iif)
    Pimreg
    Vlan2028
239.255.255.250 172.17.26.158
 pkts: 50379 bytes: 20727941 rpf failures: 0
    Vlan26 (iif)
    Pimreg
    Vlan2028
switch>
```

show ip mroute

The **show ip mroute** command displays the contents of the IP multicast routing table.

- **show ip mroute** displays information for all routes in the table.
- **show ip mroute** *gp_addr* displays information for the specified multicast group.

Command Mode

EXEC

Command Syntax

```
show ip mroute
show ip mroute gp_addr
```

Parameters

- *gp_addr* group IP address (dotted decimal notation).

Example

- This command displays the IP multicast routing table for the multicast group 225.1.1.11

```
switch>show ip mroute 225.1.1.1
PIM Sparse Mode Multicast Routing Table
Flags: E - Entry forwarding on the RPT, J - Joining to the SPT
       R - RPT bit is set, S - SPT bit is set
       W - Wildcard entry, X - External component interest
       I - SG Include Join alert rcvd, P - Ex-Prune alert rcvd
       H - Joining SPT due to policy, D - Joining SPT due to protocol
       Z - Entry marked for deletion
       A - Learned via Anycast RP Router
225.1.1.1
  172.28.1.100, 5d04h, flags: S
    Incoming interface: Vlan281
    Outgoing interface list:
      Port-Channel999
switch>
```

show ip mroute count

The **show ip mroute count** command displays IP multicast routing table statistics, including number of packets, packets per second, average packet size, and bits per second.

The **show ip mroute count** command displays the contents of the IP multicast routing table.

Command Mode

EXEC

Command Syntax

```
show ip mroute count
```

Example

- This command displays IP multicast routing table statistics.

```
switch>show ip mroute count
IP Multicast Statistics
1 groups and 1 sources
Multicast routes: 1 (*,G), 1 (S,G)
Average of 1.00 sources per group
Maximum of 1 sources per group:
    228.24.12.1
switch>
```

23.7 IGMP Commands

This section contains descriptions of the CLI commands that this chapter references.

IGMP Configuration Commands (Interface Configuration Mode)

- ip igmp last-member-query-count . . . . . Page 1026
- ip igmp last-member-query-interval . . . . . Page 1027
- ip igmp query-interval . . . . . Page 1028
- ip igmp query-max-response-time . . . . . Page 1029
- ip igmp startup-query-count . . . . . Page 1030
- ip igmp startup-query-interval . . . . . Page 1031
- ip igmp static-group . . . . . Page 1032
- ip igmp static-group acl . . . . . Page 1033
- ip igmp static-group range . . . . . Page 1034
- ip igmp version . . . . . Page 1036

IGMP Clear Commands

- clear ip igmp group . . . . . Page 1025

IGMP Display Commands

- show ip igmp groups . . . . . Page 1037
- show ip igmp interface . . . . . Page 1038
- show ip igmp static-groups . . . . . Page 1039
- show ip igmp static-groups acl . . . . . Page 1040
- show ip igmp static-groups group . . . . . Page 1041

clear ip igmp group

The **clear ip igmp group** command deletes IGMP cache entries as follows:

- **clear ip igmp group** all entries from the IGMP cache.
- **clear ip igmp group** *gp_addr* all entries for a specified multicast group.
- **clear ip igmp group interface** *int_id* all entries that include a specified interface.
- **clear ip igmp group** *gp_addr* **interface** *int_id* entries of a specified interface in a specified group.

Command Mode

Privileged EXEC

Command Syntax

clear ip igmp group [*gp_addr***] [interface *INT_ID*]**

Parameters

- *gp_addr* multicast group IP address (dotted decimal notation).
- ***INT_ID*** interface name. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Examples

- This command deletes all IGMP cache entries for the multicast group 231.23.23.14.

```
switch#clear ip igmp group 231.23.23.14
switch#
```

- This command deletes IGMP cache entries for Ethernet interface 16 in multicast group 226.45.10.45.

```
switch#clear ip igmp group 226.45.10.45 interface ethernet 16
switch#
```

ip igmp last-member-query-count

The **ip igmp last-member-query-count** command specifies the number of query messages the switch sends in response to a group-specific or group-source-specific leave message.

After receiving a message from a host leaving a group, the switch sends query messages at intervals specified by **ip igmp last-member-query-interval**. If the switch does not receive a response to the queries after sending the number of messages specified by this parameter, it stops forwarding messages to the host.

Setting the last member query count (LMQC) to 1 causes the loss of a single packet to stop traffic forwarding. While the switch can start forwarding traffic again after receiving a response to the next general query, the host may not receive that query for a period defined by **ip igmp query-interval**.

The **no ip igmp last-member-query-count** and **default ip igmp last-member-query-count** commands reset the LMQC to the default value by removing the corresponding **ip igmp last-member-query-count** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp last-member-query-count number
no ip igmp last-member-query-count
default ip igmp last-member-query-count
```

Parameters

- *number* number of query messages. Values range from 1 to 3. Default is 2.

Example

- This command configures the last-member-query-count to 3 on VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp last-member-query-count 3
switch(config-if-Vl4)#
```

ip igmp last-member-query-interval

The **ip igmp last-member-query-interval** command configures the switch's transmission interval for sending group-specific or group-source-specific query messages to the active interface.

When a switch receives a message from a host that is leaving a group it sends query messages at intervals set by this command. The **ip igmp startup-query-count** specifies the number of messages that are sent before the switch stops forwarding packets to the host.

If the switch does not receive a response after this period, it stops forwarding traffic to the host on behalf of the group, source, or channel.

The **no ip igmp last-member-query-interval** and **default ip igmp last-member-query-interval** commands reset the query interval to the default value of one second by removing the **ip igmp last-member-query-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp last-member-query-interval period
no ip igmp last-member-query-interval
default ip igmp last-member-query-interval
```

Parameters

- *period* interval (deciseconds), at which IGMP group-specific host query messages are sent.

  Value range: 10 (one second) to 317440 (8 hours, 49 minutes, 4 seconds). Default is 10 (one second).

Example

- This command configures the last-member-query-interval of 6 seconds for VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp last-member-query-interval 60
switch(config-if-Vl4)#
```

ip igmp query-interval

The **ip igmp query-interval** command configures the frequency at which the active interface, as an IGMP querier, sends host-query messages.

An IGMP querier sends query-host messages to discover the multicast groups that have members on networks attached to the interface. The switch implements a default query interval of 125 seconds.

The **no ip igmp query-interval** and **default ip igmp query-interval** commands reset the IGMP query interval to the default value of 125 seconds by removing the **ip igmp query-interval** command from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp query-interval period
no ip igmp query-interval
default ip igmp query-interval
```

Parameters

- *period* interval (seconds) between IGMP query messages. Values range from 1 to 3175 (52 minutes, 55 seconds). Default is 125.

Example

- This command configures the query-interval of 2 minutes, 30 seconds for VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp query-interval 150
switch(config-if-Vl4)#
```

ip igmp query-max-response-time

The **ip igmp query max-response-time** command configures the ***query-max-response-time*** variable for the configuration mode interface. This variable is used to set the Max Response Time field in outbound Membership Query messages. Max Response Time specifies the maximum period a recipient can wait before responding with a Membership Report.

The router with the lowest IP address on a subnet sends membership queries as the IGMP querier. When a router receives a membership query from a source with a lower IP address, it resets its query timer. Upon timer expiry, the router begins sending membership queries. If the router subsequently receives a membership query from a router with a lower IP address, it stops sending membership queries and resets the query maximum response timer.

The **no ip igmp query-max-response-time** and **default ip igmp query-max-response-time** commands restore the default query-max-response-time of 10 seconds for the configuration mode interface by removing the corresponding the **ip igmp query max-response-time** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp query-max-response-time period
no ip igmp query-max-response-time
default ip igmp query-max-response-time
```

Parameters

- *period* maximum response time (deciseconds). Values range from 1 to 31744 (52 minutes, 54 seconds). Default is 100 (ten seconds).

Example

- This command configures the query-max-response-time of 18 seconds for VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp query-max-response-time 180
switch(config-if-Vl4)#
```

ip igmp startup-query-count

The **ip igmp startup-query-count** command specifies the number of query messages that an interface sends during the startup interval defined by **ip igmp startup-query-interval**.

When an interface starts running IGMP, it can establish the group state more quicker by sending query messages at a higher frequency. The ***startup-query-interval*** and ***startup-query-count*** parameters define the startup period and the query message transmission frequency during that period.

The **no ip igmp startup-query-count** and **default ip igmp startup-query-count** commands restore the default ***startup-query-count*** value of 2 for the configuration mode interface by removing the corresponding **ip igmp startup-query-count** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp startup-query-count number
no ip igmp startup-query-count
default ip igmp startup-query-count
```

Parameters

- *number* number of queries. Values range from 1 to 65535. Default is 2.

Example

- This command configures the startup query count of 10 for VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp startup-query-count 10
switch(config-if-Vl4)#
```

ip igmp startup-query-interval

The **ip igmp startup-query-interval** command specifies the configuration mode interface's IGMP startup period, during which query messages are sent at an accelerated rate.

When an interface starts running IGMP, it can establish the group state quicker by sending query messages at a higher frequency. The ***startup-query-interval*** and ***startup-query-count*** parameters define the startup period and the query message transmission frequency during that period.

The **no ip igmp startup-query-interval** and **default ip igmp startup-query-interval** commands restore the configuration mode interface's default IGMP ***startup-query-interval*** of 31 seconds by removing the corresponding **ip igmp startup-query-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp startup-query-interval period
no ip igmp startup-query-interval
default ip igmp startup-query-interval
```

Parameters

- *period* startup query interval, in deciseconds. Values from 10 (one second) to 317440 (8 hours, 49 minutes, 4 seconds). Default is 31 seconds.

Example

- This command configures the startup query count of one minute for VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp startup-query-interval 600
switch(config-if-Vl4)#
```

ip igmp static-group

The **ip igmp static-group** command configures the configuration mode interface as a static member of a specified multicast group. This allows the router to forward multicast group packets through the interface without otherwise appearing or acting as a group member. By default, static group memberships are not configured on any interfaces.

If the command includes a source address, only multicast group messages received from the specified host address are fast-switched. Otherwise, all multicast messages of the specified group are fast-switched.

The **no ip igmp static-group** and **default ip igmp static-group** commands removes the configuration mode interface's static group membership command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp static-group group_address [SOURCE_ADDRESS]
no ip igmp static-group group_address [SOURCE_ADDRESS]
default ip igmp static-group group_address [SOURCE_ADDRESS]
```

Parameters

- *group_address* IPv4 address of multicast group for which the interface fast-switches packets.
- ***SOURCE_ADDRESS*** IP address of host that originates multicast data packets.
  - <no parameter> all multicast messages of the specified group are fast-switched.
  - *ipv4_address* source IP address (dotted decimal notation).

Related Commands

- **ip igmp static-group acl** command configures the configuration mode interface as a static member of the multicast groups specified by an IP access control list (ACL).
- **ip igmp static-group range** command configures the configuration mode interface as a static member of multicast groups specified by an address range. A single **ip igmp static-group range** command is the equivalent of multiple **ip igmp static-group** commands

Example

- This command configures VLAN interface 4 as a static member of the multicast group 241.1.1.45 for data packets that originate at 15.1.1.1.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp static-group 241.1.1.45 15.1.1.1
switch(config-if-Vl4)#
```

ip igmp static-group acl

The **ip igmp static-group acl** command configures the configuration mode interface as a static member of the multicast groups specified by an IP access control list (ACL). This command is a variant of the **ip igmp static-group** command that uses ACL rules to specify a set of source-multicast group address pairs instead of specifying a single pair. Multiple static-group ACLs can be assigned to an interface. Static groups can be assigned manually and through ACLs simultaneously.

Access control lists that this command references must contain rules of the following format.

- **permit <protocol><source><destination>**, where
  - **<protocol>** has no effect on the static group.
  - **<source>** address of host originating multicast data packets. Must be a host address.
  - **<destination>** multicast group IP address. Must be a multicast address, may be a subnet.

An ACL can contain multiple rules. An ACL can be applied to an interface only when all of its rules comply to the specified restrictions. The **show ip igmp static-groups acl** displays the source-multicast group pairs that the specified list configures and lists issues with illegal rules.

The **no ip igmp static-group acl** and **default ip igmp static-group acl** commands remove the specified static group ACL command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp static-group acl ACL_NAME
no ip igmp static-group acl ACL_NAME
default ip igmp static-group acl ACL_NAME
```

Parameters

- *ACL_NAME* access control list that specifies the multicast group addresses for which the interface fast-switches packets.

Example

- This command configures VLAN interface 4 as a static member of the multicast group specified by the ACL named LIST_1.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp static-group acl LIST_1
switch(config-if-Vl4)#
```

ip igmp static-group range

The **ip igmp static-group range** command configures the configuration mode interface as a static member of multicast groups specified by an address range. This allows the router to forward multicast group packets through the interface without otherwise appearing or acting as a group member. By default, no static group memberships are configured on interfaces.

This command is a variant of the **ip igmp static-group** command that allows the assignment of a subnet range of source addresses or a subnet range of multicast groups. A single **ip igmp static-group range** command is the equivalent of multiple **ip igmp static-group** commands, each of which can only assign a single multigroup-source pair to an interface. Running-config converts the range command to the equivalent list of **ip igmp static-group** commands.

If the command includes a source address range, only multicast group messages received from the range are fast-switched. Otherwise, all multicast messages of the specified group are fast-switched.

The **no ip igmp static-group range** and **default ip igmp static-group range** commands remove the specified range of static group statements from ***running-config***. The **no ip igmp static-group** and **default ip igmp static-group** commands can remove an individual static-group command that was initially added to ***running-config*** by an **ip igmp static-group range** command.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp static-group range GROUP_ADDR [SOURCE_ADDR]
no ip igmp static-group range GROUP_ADDR [SOURCE_ADDR]
default ip igmp static-group range GROUP_ADDR [SOURCE_ADDR]
```

Parameters

- ***GROUP_ADDR*** address of multicast group for which the interface fast-switches packets.
  - *gp_ipv4_addr* multicast group IPv4 address.
  - *gp_ipv4_subnet* IPv4 subnet address of multicast groups (CIDR or address-mask).
- ***SOURCE_ADDR*** IP address of a host range that originates multicast data packets.
  - <no parameter> all multicast messages of the specified range are fast-switched.
  - **source** *sr_ipv4_address* source IPv4 address (dotted decimal notation).
  - **source** *sr_ ipv4_subnet* IPv4 subnet address of source hosts (CIDR or address- mask).

Warning A command cannot specify a subnet address for both multicast group and source.

Examples

- This command configures VLAN interface 4 as a static member of the multicast group range 241.1.4.1/24 for data packets that originate at 15.1.1.1.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp static-group range 239.1.4.1/24 source 15.1.1.1
switch(config-if-Vl4)#
```

- This command attempts to configure VLAN interface 4 as a static member of the multicast group range 241.1.4.1/24 for data packets that originate at the 15.1.1.1/29 subnet. Because the range and source cannot both be subnets, this command generates an error message.

```
switch(config-if-Vl4)#ip igmp static-group range 239.1.1.1/29 source 16.1.1.1/29
% Error: cannot specify source range with group range
switch(config-if-Vl4)#
```

ip igmp version

The **ip igmp version** command configures the Internet Group Management Protocol (IGMP) version on the configuration mode interface. Version 3 is the default IGMP version.

IGMP is enabled by the **ip pim sparse-mode** command. The ig igmp version command does not effect the IGMP enabled status.

The **no ip igmp version** and **default ip igmp version** commands restores the configuration mode interface to IGMP version 3 by removing the **ip igmp version** statement from ***running-config.***

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip igmp version version_number
no ip igmp version
default ip igmp version
```

Parameters

- *version_number* IGMP version number. Value ranges from 1 to 3.

Example

- This command configures IGMP version 3 on VLAN interface 4

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip igmp version 3
switch(config-if-Vl4)#
```

show ip igmp groups

The **show ip igmp groups** command displays multicast groups that have receivers directly connected to the switch, as learned through Internet Group Management Protocol (IGMP).

- **show ip igmp groups** all multicast groups.
- **show ip igmp groups** *group_addr* listed multicast group.
- **show ip igmp groups interface** *int_name* all multicast groups on specified interfaces
- **show ip igmp groups** *group_addr* **interface** *int_name* listed multicast group on specified interface.

Command Mode

EXEC

Command Syntax

show ip igmp groups *GROUP_LIST* [*DATA*]

Parameters

- ***GROUP_LIST*** list of groups for which the command displays information. . Options include:
  - <no parameter> all multicast groups.
  - *group_addr* single multicast group address (dotted decimal notation).
  - **interface ethernet** *e_num* all multicast groups on Ethernet interface (*e_num*).
  - **interface loopback** *l_num* all multicast groups on Loopback interface (*l_num*).
  - **interface management** *m_num* all multicast groups on Management interface (*m_num*).
  - **interface port-channel** *p_num* all multicast groups on Port-Channel Interface (*p_num*).
  - **interface vlan** *v_num* all multicast groups on VLAN interface (*v_num*).
- ***DATA*** specifies the type of information displayed. Options include
  - <no parameter> proivdes uptime, expiration, and address of reporter.
  - **detail** also include group mode and group source list.

Example

- This command displays multicast groups with receivers directly connected to the switch.

```
switch>show ip igmp groups
NOTE: static-group information not shown below.  Use the
      'show ip igmp static-groups' command.
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
225.12.1.1       Vlan162                  11d01h    00:02:57  172.17.2.110
225.12.1.2       Vlan162                  11d01h    00:02:57  172.17.2.110
225.12.1.3       Vlan162                  11d01h    00:02:57  172.17.2.110
225.12.1.4       Vlan162                  11d01h    00:02:57  172.17.2.110
225.12.1.5       Vlan162                  11d01h    00:02:57  172.17.2.110
switch>
```

show ip igmp interface

The **show ip igmp interface** command displays multicast-related information about an interface.

- **show ip igmp interface** – displays all multicast information for all interfaces
- **show ip igmp interface** *int-name* – displays multicast information for the specified interfaces.

When all arguments are omitted, the command displays information for all interfaces.

Command Mode

EXEC

Command Syntax

show ip igmp interface [***INT_NAME***]

Parameters

- ***INT_NAME*** Interface type and number. Values include
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command displays multicast related information about VLAN 26.

```
switch>show ip igmp interface vlan 26
Vlan26 is up
  Interface address: 172.17.26.1/23
  IGMP on this interface: enabled
  Multicast routing on this interface: enabled
  Multicast TTL threshold: 1
  Current IGMP router version: 2
  IGMP query interval: 125 seconds
  IGMP max query response time: 100 deciseconds
  Last member query response interval: 10 deciseconds
  Last member query response count: 2
  IGMP querier: 172.17.26.1
  Robustness: 2
  Require router alert: enabled
  Startup query interval: 312 deciseconds
  Startup query count: 2
  General query timer expiry: 00:00:22
  Multicast groups joined:
    239.255.255.250

switch>
```

show ip igmp static-groups

The **show ip igmp static-groups** command displays information about all configured IGMP multicast static groups. IGMP multicast static groups are assigned with the **ip igmp static-group** command.

Command Mode

EXEC

Command Syntax

show ip igmp static-groups [*INFO_LEVEL*] [interface *INT_NAME*]

Parameters

- *INFO_LEVEL* specifies the type of information displayed. Options include
  - <no parameter> VLAN number and port-list for each group.
  - **detail** port-specific information for each group, including transmission times and expiration.
- *INT_NAME* Interface type and number. Values include
  - <no parameter> static groups on all interfaces.
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Related Commands

- **show ip igmp static-groups acl**
- **show ip igmp static-groups group**

Examples

- This command displays information about all multicast static groups.

```
switch>show ip igmp static-groups
Interface Vlan281:
        Manually configured groups:
Interface Port-Channel999:
        Manually configured groups:
switch>
```

- This command displays information about the multicast static groups on VLAN interface 21.

```
switch>show ip igmp static-groups interface vlan 21
Interface Vlan281:
        Manually configured groups:
switch>
```

show ip igmp static-groups acl

The **show ip igmp static-groups acl** command displays information about the IGMP multicast static groups that are configured by the specified access control list (ACL). The command also displays problems with an ACL that prevent its assignment to an interface.

Command Mode

EXEC

Command Syntax

show ip igmp static-groups acl

Example

The following **show ip igmp static-group acl** command example references these ACLs:

```
ip access-list 1
   10 permit igmp host 10.1.1.1 225.1.1.0/29
   20 permit igmp host 10.1.1.2 225.1.1.0/29
!
ip access-list 2
   10 permit igmp 10.1.1.0/29 host 225.1.1.1
!
ip access-list 3
   10 deny igmp host 10.1.1.1 255.1.1.0/29
!
ip access-list 4
   10 permit igmp host 10.1.1.1 225.1.1.0/29
   20 permit igmp 10.1.1.0/29 host 225.1.1.1
```

- This command displays static group configuration data about the various ACLs.

```
switch>show ip igmp static-group acl 1
acl 1
        ( 10.1.1.1, 225.1.1.0/29 )
        ( 10.1.1.2, 225.1.1.0/29 )
Interfaces using this ACL for static groups:
        Ethernet12
switch>show ip igmp static-group acl 2
acl 2
        Seq no 30: source address must be a single host or *, not a range
Interfaces using this ACL for static groups:
        Ethernet8
switch>show ip igmp static-group acl 3
acl 4
        Seq no 10: action must be 'permit'
Interfaces using this ACL for static groups:
         none
switch>show ip igmp static-group acl 4
acl 5
        ( 10.1.1.1, 225.1.1.0/29 )
        Seq no 20: source address must be a single host or *, not a range
Interfaces using this ACL for static groups:
         none
switch>
```

show ip igmp static-groups group

The **show ip igmp static-groups group** command displays information about all specified IGMP multicast static groups. IGMP multicast static groups are assigned with the **ip igmp static-group** command.

Command Mode

EXEC

Command Syntax

```
show ip igmp static-groups group [GROUP_LIST]
```

Parameters

- *GROUP LIST* Groups for which command displays information
  - <no parameter> all multicast groups.
  - *group_address* single multicast group address (dotted decimal notation).

Related Commands

- **show ip igmp static-groups**

23.8 IGMP Snooping Commands

This section contains descriptions of the CLI commands that this chapter references.

IGMP Snooping Configuration Commands (Global)

- ip igmp profile. . . . . Page 1044
- ip igmp snooping . . . . . Page 1046
- ip igmp snooping immediate-leave. . . . . Page 1048
- ip igmp snooping querier . . . . . Page 1049
- ip igmp snooping querier address. . . . . Page 1050
- ip igmp snooping querier max-response-time . . . . . Page 1051
- ip igmp snooping querier query-interval. . . . . Page 1052
- ip igmp snooping robustness-variable . . . . . Page 1053
- ip igmp snooping vlan. . . . . Page 1054
- ip igmp snooping vlan max-groups . . . . . Page 1055
- ip igmp snooping vlan mrouter. . . . . Page 1056
- ip igmp snooping vlan querier. . . . . Page 1057
- ip igmp snooping vlan querier address . . . . . Page 1059
- ip igmp snooping vlan querier max-response-time . . . . . Page 1060
- ip igmp snooping vlan querier query-interval . . . . . Page 1061
- ip igmp snooping vlan static. . . . . Page 1062

IGMP Configuration Commands (Interface Configuration Mode)

- ip igmp snooping filter . . . . . Page 1047

IGMP Snooping Clear Commands

- clear ip igmp snooping counters . . . . . Page 1043

IGMP Snooping Display Commands

- show ip igmp profile . . . . . Page 1065
- show ip igmp snooping. . . . . Page 1066
- show ip igmp snooping counters . . . . . Page 1067
- show ip igmp snooping counters ethdev-pams . . . . . Page 1068
- show ip igmp snooping groups. . . . . Page 1069
- show ip igmp snooping groups count . . . . . Page 1072
- show ip igmp snooping mrouter. . . . . Page 1073
- show ip igmp snooping querier. . . . . Page 1074

IGMP Profile Configuration Mode Commands

- permit / deny. . . . . Page 1063
- range. . . . . Page 1064

clear ip igmp snooping counters

The **clear ip igmp snooping counters** command resets the snooping message counters for the specified interface. The snooping counters for all interfaces are reset if the command does include an interface name.

The **show ip igmp snooping counters** command displays the counter contents. See the **show ip igmp snooping counters** command description for a list of available snooping counters.

Command Mode

Global Configuration

Command Syntax

```
clear ip igmp snooping counters [INT_NAME]
```

Parameters

- *INT_NAME* interface name. Formats include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num.*
  - **port-channel** *p_num* Port-channel interface specified by *p_num*.
  - **switch** virtual interface to an L2 querier.

Example

- This command clears the snooping counters for messages received on Ethernet interface 15.

```
switch(config)#clear ip igmp snooping counters ethernet 15
switch(config)#
```

ip igmp profile

The **ip igmp profile** command places the switch in IGMP-profile configuration mode to configure an IGMP profile. IGMP profiles control the multicast groups that an interface can join.

Profiles consist of the filter type and an address range:

- Filter types specify accessibility to the listed address range:
  - Permit filters define the multicast groups the interface can join.
  - Deny filters define the multicast groups the interface cannot join.

  Profiles are deny filters by default.
- Address ranges specify a list of addresses and ranges:
  - In permit filters, the permitted groups are specified by the address range.
  - In deny filters, all groups are permitted except those specified by the address range.

Implementing IGMP filtering affects IGMP report forwarding as follows:

- IGMPv2: The report is dropped for unallowed multicast groups and forwarded to mrouters for permitted groups.
- IGMPv3: There may be multiple group records in a report.
  - No groups are allowed: The report is dropped.
  - All groups are allowed: The report is forwarded to mrouter ports as normal.
  - Some groups are allowed: A revised report is forwarded to mrouter ports.

    The revised report includes records for the allowed group addresses with the same source MAC and IP addresses.

The **no ip igmp profile** and **default ip igmp profile** commands delete the specified IGMP profile from *running-config*.

IGMP-profile configuration mode is not a group change mode; ***running-config*** is changed immediately upon entering commands. Exiting IGMP-profile configuration mode does not affect the configuration.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
ip igmp profile profile_name
no ip igmp profile profile_name
default ip igmp profile profile_name
```

Parameters

- *profile_name* name of the IGMP profile.

Commands Available in igmp-profile Configuration Mode

- **permit / deny**
- **range**

Related Commands

- **ip igmp snooping filter** applies an IGMP snooping filter to a configuration mode interface.

Example

- These commands enter IGMP-profile configuration mode and configure the profile as a permit list.

```
switch(config)#ip igmp profile list_1
switch(config-igmp-profile-list_1)#permit
switch(config-igmp-profile-list_1)#
```

ip igmp snooping

The **ip igmp snooping** command enables snooping globally. By default, global snooping is enabled.

When global snooping is enabled, **ip igmp snooping vlan** enables or disables snooping on individual VLANs. When global snooping is disabled, snooping cannot be enabled on individual VLANs.

QoS does not support IGMP packets when IGMP snooping is enabled.

The **no ip igmp snooping** command disables global snooping. The **default ip igmp snooping** command restores the global snooping default setting of enabled by removing the **ip igmp snooping** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping
no ip igmp snooping
default ip igmp snooping
```

Example

- This command globally enables snooping on the switch.

```
switch(config)#ip igmp snooping
switch(config)#
```

ip igmp snooping filter

The **ip igmp snooping filter** command applies the specified IGMP snooping profile to the configuration mode interface. An IGMP snooping profile specifies the multicast groups that an interface may join. Profiles consists of the filter type and an address range:

- Filter type: Specifies accessibility to the listed address range:
  - Permit filters define the multicast groups the interface can join.
  - Deny filters define the multicast groups the interface cannot join.
- Address range: Specifies a list of addresses and ranges.
  - In permit filters, the permitted groups are specified by the address range.
  - In deny filters, all groups are permitted except those specified by the address range.

An interface without a snooping profile assignment may join any multicast group.

Snooping profiles are configured in IGMP-profile configuration mode (**ip igmp profile**).

The **no ip igmp snooping filter** and **default ip igmp snooping filter** commands restore the default setting of allowing an interface to join any multicast group by deleting the corresponding **ip igmp snooping filter** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
ip igmp snooping filter profile_name
no ip igmp snooping filter [profile_name]
default ip igmp snooping filter [profile_name]
```

Parameters

- *profile_name* name of profile assigned to interface.

Example

- This command applies the ***list_1*** snooping profile to Ethernet interface 7.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#ip igmp snooping filter list_1
switch(config-if-Et7)#
```

ip igmp snooping immediate-leave

The **ip igmp snooping vlan immediate-leave** command enables fast-leave processing on specified VLANs. When fast-leave processing is enabled, the switch immediately removes a VLAN from the multicast group when it detects an IGMP version 2 leave message on that VLAN. IGMP fast-leave processing is enabled on all VLANs by default.

The **no ip igmp snooping vlan immediate-leave** command disables fast-leave processing on the specified VLANs. The **default ip igmp snooping vlan immediate-leave** command restores fast-leave processing on the specified VLANs by removing the corresponding **no ip igmp snooping vlan immediate-leave** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range immediate-leave
no ip igmp snooping vlan v_range immediate-leave
default ip igmp snooping vlan v_range immediate-leave
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.

Example

- This command enables IGMP fast-leave processing on VLAN 10.

```
switch(config)#ip igmp snooping vlan 10 immediate-leave
switch(config)#
```

ip igmp snooping querier

The **ip igmp snooping querier** command enables the snooping querier globally, which control the querier for VLANs with no snooping querier command. The **ip igmp snooping vlan querier** controls the querier on individual VLANs.

The IGMP snooping querier supports snooping by sending layer 2 membership queries to hosts attached to the switch. The snooping querier is enabled when snooping is enabled or PIM is not enabled on the switch. The IGMP snooping querier performs these actions when enabled:

- Remains idle until it detects IGMP traffic from a multicast router.
- Starts when it does not detect IGMP traffic for 60 seconds.
- Quits when it detects IGMP traffic from a multicast router.

The **no ip igmp snooping querier** and **default ip igmp snooping querier** commands disable the snooping querier globally by removing the **ip igmp snooping querier** statement from ***running-config***. The snooping querier is globally disabled by default.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier
no ip igmp snooping querier
default ip igmp snooping querier
```

Example

- This commands globally enable the snooping querier on the switch.

```
switch(config)#ip igmp snooping querier
switch(config)#
```

ip igmp snooping querier address

The **ip igmp snooping querier address** command sets the global querier source IP address, which specifies the source address for packets transmitted from VLANs for which a querier address (**ip igmp snooping vlan querier address**) is not configured. To use a snooping querier, an address must be explicitly configured globally or for the VLAN.

The switch does not define a default global querier address.

The **no ip igmp snooping querier address** and **default ip igmp snooping querier address** commands remove the global querier address command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier address ipv4_address
no ip igmp snooping querier address
default ip igmp snooping querier address
```

Parameters

- *ipv4_address* source IPv4 address.

Example

- This command sets the source IP address to 10.1.1.41 for query packets transmitted from the switch.

```
switch(config)#ip igmp snooping querier address 10.1.1.41
switch(config)#
```

ip igmp snooping querier max-response-time

The **ip igmp snooping querier max-response-time** command specifies the global ***max-response-time*** value. The switch uses ***max-response-time*** to set the Max Response Time field in outbound Membership Query messages. Max Response Time specifies the maximum period a recipient can wait before responding with a Membership Report.

VLANs not assigned a ***max-response-time*** value (**ip igmp snooping vlan querier max-response-time**) use the global value. VLAN commands take precedence over the global value.

Values range from 1 to 25 seconds. The default global value is 10 seconds.

The **no ip igmp snooping querier max-response-time** and **default ip igmp snooping querier max-response-time** commands restore the global ***max-response-time*** default value by removing the **ip igmp snooping querier max-response-time** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier max-response-time resp_sec
no ip igmp snooping querier max-response-time
default ip igmp snooping querier max-response-time
```

Parameters

- *resp_sec* ***max-response-time*** value (seconds). Values range from 1 to 25. Default (global) is 10.

Example

- This command sets the global max-response-time to 15 seconds.

```
switch(config)#ip igmp snooping querier max-response-time 15
switch(config)#
```

ip igmp snooping querier query-interval

The **ip igmp snooping querier query-interval** command sets the global query interval. Values range from 5 to 3600 seconds. The default global value is 125 seconds. The query interval is the period between IGMP Membership Query messages sent from a snooper querier. The global value specifies the query interval for VLANs with no query-interval command.

VLANs not assigned a ***query interval*** value (**ip igmp snooping vlan querier query-interval**) use the global value. VLAN commands take precedence over the global value.

The **no ip igmp snooping querier query-interval** and **default ip igmp snooping querier query-interval** commands reset the global query-interval value to 125 seconds by removing the **ip igmp snooping querier query-interval** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping querier query-interval query_sec
no ip igmp snooping querier query-interval
default ip igmp snooping querier query-interval
```

Parameters

- *query_sec* query interval (seconds). Values range from 5 to 3600. Default (global) is 125.

Example

- This command sets the global query interval to 150 seconds.

```
switch(config)#ip igmp snooping querier query-interval 150
switch(config)#
```

ip igmp snooping robustness-variable

The **ip igmp snooping robustness-variable** command configures the robustness variable for snooping packets sent from any VLAN. Values range from 1 to 3 with a default of 2.

The robustness variable specifies the number of unacknowledged snooping queries that a switch sends before removing the recipient from the group list.

The **no ip igmp snooping robustness-variabl**e and **default ip igmp snooping robustness-variabl**e commands reset the robustness variable to 2 by removing the **ip igmp snooping robustness-variable** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping robustness-variable robust_value
no ip igmp snooping robustness-variable
default ip igmp snooping robustness-variable
```

Parameters

- *robust_value* robustness variable. Values range from 1 to 3. Default is 2.

Example

- This command sets the robustness-variable value to 3.

```
switch(config)#ip igmp snooping robustness-variable 3
switch(config)#
```

ip igmp snooping vlan

The **ip igmp snooping vlan** command enables snooping on individual VLANs if snooping is globally enabled. By default, IGMP snooping is enabled on all VLANs. The **ip igmp snooping** command enables snooping globally.

QoS does not support IGMP packets when IGMP snooping is enabled.

The **no ip igmp snooping vlan** command disables snooping on the specified VLANs.

The **default ip igmp snooping vlan** command returns the snooping setting for the specified VLANs to enabled by removing the corresponding **ip igmp snooping vlan** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range
no ip igmp snooping vlan v_range
default ip igmp snooping vlan v_range
```

Parameters

- *v_range* VLANs upon which snooping is enabled. Formats include a number, a number range, or a comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.

Example

- This command disables snooping on VLANs 2 through 4.

```
switch(config)#no ip igmp snooping vlan 2-4
switch(config)#
```

ip igmp snooping vlan max-groups

The **ip igmp snooping vlan max-groups** command specifies the number of multicast groups that the active VLAN's forwarding table can contain. After the limit is reached, attempts to join new groups are ignored. By default, there is no limit to the number of groups.

The **no ip igmp snooping vlan max-groups** and **default ip igmp snooping vlan max-groups** removes the maximum group limit by deleting the **ip igmp snooping vlan max-groups** statement from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range max-groups quantity
no ip igmp snooping vlan v_range max-groups
default ip igmp snooping vlan v_range max-groups
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *quantity* maximum number of multicast groups that can access the interface. Values range from 0 to 65534.

Examples

- This command limits the number of multicast groups that hosts on VLAN 6 can simultaneously access to 25.

```
switch(config)#ip igmp snooping vlan 6 max-groups 25
switch(config)#
```

- This command allows each VLAN between 8 and 15 to receive multicast packets from 30 groups.

```
switch(config)#ip igmp snooping vlan 8-15 max-groups 30
switch(config)#
```

- This command removes the maximum group restriction from all VLAN interfaces between 1 and 50.

```
switch(config)#no ip igmp snooping vlan 1-50 max-groups
switch(config)#
```

ip igmp snooping vlan mrouter

The **ip igmp snooping vlan mrouter** command adds a multicast router as a static port to the specified VLANs. The router port must be in the specified VLAN range.

Snooping may not always be able to locate the IGMP querier. This command should specify IGMP queriers that are known to connect to the network through a port on the switch.

The **no ip igmp snooping vlan mrouter** and **default ip igmp snooping vlan mrouter** commands remove the specified static port configuration by deleting the corresponding **ip igmp snooping vlan mrouter** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range mrouter interface STATIC_INT
no ip igmp snooping vlan v_range mrouter interface STATIC_INT
default ip igmp snooping vlan v_range mrouter interface STATIC_INT
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- ***STATIC_INT*** interface the command configures as a static port. Selection options include:
  - **ethernet** *e_range* where *e_range* is the number, range, or list of ethernet ports
  - **port-channel** *p_range* where *p_range* is the number, range, or list of channel ports

  The ***STATIC_INT*** interface must route traffic through a VLAN specified within *v_range*.

Example

- This command configures the static connection to a multicast router through Ethernet port 3.

```
switch(config)#ip igmp snooping vlan 2 mrouter interface ethernet 3
switch(config)#
```

ip igmp snooping vlan querier

The **ip igmp snooping vlan querier** command controls the querier for the specified VLANs. VLANs follow the global querier setting unless overridden by one of these commands:

- **ip igmp snooping vlan querier** enables the querier on specified VLANs.
- **no ip igmp snooping vlan querier** disables the querier on specified VLANs.

VLAN querier commands take precedence over the global querier setting. The **ip igmp snooping querier** controls the querier for VLANs with no snooping querier command.

The IGMP snooping querier supports snooping by sending layer 2 membership queries to hosts attached to the switch. The snooping querier is enabled when snooping is enabled or PIM is not enabled on the switch. The IGMP snooping querier performs these actions when enabled:

- Remains idle until it detects IGMP traffic from a multicast router.
- Starts when it does not detect IGMP traffic for 60 seconds.
- Quits when it detects IGMP traffic from a multicast router.

The **default ip igmp snooping vlan querier** command restores the usage of the global setting for the specified VLAN by removing the corresponding **ip igmp snooping vlan querier** or **no ip igmp snooping vlan querier** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier
no ip igmp snooping vlan v_range querier
default ip igmp snooping vlan v_range querier
```

Parameters

- *v_range* VLAN IDs. Formats include a number, a number range, or a comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.

Examples

- These commands globally enable the snooping querier on the switch, explicitly disable snooping on VLANs 1-3, and explicitly enable snooping on VLANs 4-6.

```
switch(config)#ip igmp snooping querier
switch(config)#no ip igmp snooping vlan 1-3 querier
switch(config)#ip igmp snooping vlan 4-6 querier
```

After running these commands, the running-config file contains these lines, which indicate that the snooping querier is enabled on VLANs 4-6.

```
switch(config)#show running-config

<-------OUTPUT OMITTED FROM EXAMPLE-------->
no ip igmp snooping vlan 1 querier
no ip igmp snooping vlan 2 querier
no ip igmp snooping vlan 3 querier
ip igmp snooping vlan 4 querier
ip igmp snooping vlan 5 querier
ip igmp snooping vlan 6 querier
ip igmp snooping querier
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

- This command removes the querier setting for VLANs 2-5:

```
switch(config)#default ip igmp snooping vlan 2-5 querier
```

When executed after the previous commands, the snooping querier is disabled explicitly on VLANs 1-2, enabled implicitly on VLANs 3-6, and enabled explicitly on VLANs 7-8, as shown by ***running-config***:

```
<-------OUTPUT OMITTED FROM EXAMPLE-------->
no ip igmp snooping vlan 1 querier
ip igmp snooping vlan 6 querier
ip igmp snooping querier
<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

- This command sets the global snooping querier to disabled by removing the global querier setting from ***running-config***:

```
switch(config)#no ip igmp snooping querier
switch(config)#
```

When executed after the previous commands, the snooping querier is disabled explicitly on VLANs 1-2, disabled implicitly on VLANs 3-6 and enabled explicitly on VLANs 7-8, as shown by ***running-config***.

```
<-------OUTPUT OMITTED FROM EXAMPLE-------->
no ip igmp snooping vlan 1 querier
ip igmp snooping vlan 6 querier

<-------OUTPUT OMITTED FROM EXAMPLE-------->
```

ip igmp snooping vlan querier address

The **ip igmp snooping vlan querier address** command sets the source address for query packets sent from specified VLANs. VLANs not assigned an address use the global address (**ip igmp snooping querier address**). VLAN querier address commands take precedence over the global address.

To use a snooping querier, an address must be explicitly configured globally or for the querier's VLAN.

The **no ip igmp vlan snooping querier address** and **default ip igmp snooping vlan querier address** commands reset the specified VLAN to use the global address by removing the corresponding **ip igmp snooping vlan querier address** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier address ipv4_address
no ip igmp snooping vlan v_range querier address
default ip igmp snooping vlan v_range querier address
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *ipv4_address* source IPv4 address.

Example

- This command sets the source IPv4 address of 10.14.1.1. for query packets transmitted from VLAN 2.

```
switch(config)#ip igmp snooping vlan 2 querier address 10.14.1.1
switch(config)#
```

ip igmp snooping vlan querier max-response-time

The **ip igmp snooping vlan querier max-response-time** command configures ***max-response-time*** for packets sent from the specified VLANs. VLANs not assigned a value use the global setting (**ip igmp snooping querier max-response-time**). VLAN commands take precedence over the global value.

Switches use ***max-response-time*** to set the Max Response Time field in outbound Membership Query messages. Max Response Time specifies the maximum period a recipient can wait before responding with a Membership Report.

The **no ip igmp snooping vlan querier max-response-time** and **default ip igmp snooping vlan querier max-response-time** commands reset the specified VLAN to use the global ***max-response-time*** by removing the corresponding **ip igmp snooping vlan querier max-response-time** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier max-response-time resp_sec
no ip igmp snooping vlan v_range querier max-response-time
default ip igmp snooping vlan v_range querier max-response-time
```

Parameters

- *v_range* VLAN ID. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *resp_sec* ***max-response-time*** value (seconds). Values range from 1 to 25. Default (global) is 10.

Example

- This command sets the max-response-time for VLAN 2 to 5 seconds.

```
switch(config)#ip igmp snooping vlan 2 querier max-response-time 5
switch(config)#
```

ip igmp snooping vlan querier query-interval

The **ip igmp snooping vlan querier query-interval** command sets the query interval for the specified VLAN. VLANs not assigned a value use the global value (**ip igmp snooping querier query-interval**). VLAN commands have precedence over the global value. The query interval is the period between IGMP Membership Query messages sent from a snooper querier

The **no ip igmp snooping vlan querier query-interval** and **default ip igmp snooping vlan querier query-interval** commands reset the specified VLAN to use the global value by removing the corresponding **ip igmp snooping vlan querier query-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_range querier query-interval query_sec
no ip igmp snooping vlan v_range querier query-interval
default ip igmp snooping vlan v_range querier query-interval
```

Parameters

- *v_range* VLAN IDs. Formats include a number, number range, or comma-delimited list of numbers and ranges. Numbers range from 1 to 4094.
- *query_sec* query interval (seconds). Values range from 5 to 3600. Default (global) is 125.

Example

- This command sets the query interval for VLAN 10 to 240 seconds.

```
switch(config)#ip igmp snooping vlan 10 querier query-interval 240
switch(config)#
```

ip igmp snooping vlan static

The **ip igmp snooping static** command adds ports as static members to a multicast group. The ports must be in the specified VLAN range.

The **no ip igmp snooping static** and **default ip igmp snooping static** commands remove the specified ports from the multicast group by deleting the corresponding ip igmp snooping static statements from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip igmp snooping vlan v_num static ipv4_addr interface STATIC_INT
no ip igmp snooping vlan v_num static ipv4_addr interface STATIC_INT
default ip igmp snooping vlan v_num static ipv4_addr interface STATIC_INT
```

Parameters

- *v_num* VLAN number. Value ranges from 1 to 4094.
- *ipv4_addr* multicast group IPv4 address.
- ***STATIC_INT*** interface the command configures as the static group member. Options include:
  - **ethernet** *e_range*, where *e_range* is the number, range, or list of Ethernet ports
  - **port-channel** *p_range*, where *p_range* is the number, range, or list of channel ports

Example

- This command configures the static connection from VLAN 2 to the multicast group at 224.2.1.4 through Ethernet port 3.

```
switch(config)#ip igmp snooping vlan 2 static 224.2.1.4 interface ethernet 3
switch(config)#
```

permit / deny

The **permit** command configures the configuration mode IGMP profile as a permit list. Applying a permit list to an interface restricts that interface from joining any multicast group not included in the list.

IGMP profiles are deny lists by default. When applied to an interface, a deny list allows the interface to join any multicast group that is not included in the list.

The **deny** command restores the IGMP list to its default type by removing the corresponding **permit** statement from ***running-config***.

The **range** command adds and removes address ranges from the configuration mode profile.

Command Mode

IGMP-profile Configuration

Command Syntax

```
permit
deny
```

Related Commands

- **ip igmp profile** places the switch in IGMP-profile configuration mode.

Example

- These commands enter IGMP profile configuration mode and configure the profile as a permit list.

```
switch(config)#ip igmp profile list_1
switch(config-igmp-profile-list_1)#permit
switch(config-igmp-profile-list_1)#
```

range

The **range** command specifies an address range for the configuration mode IGMP profile. A permit range specifies the groups that an interface is permitted to join. A deny range specifies the groups that an interface is not permitted to join. The **permit / deny** command specifies the range type.

A profile may contain multiple range statements to define a discontiguous address range.

The **no range** and **default range** commands remove the specified address range from a previous specified list.

Command Mode

IGMP-Profile Configuration

Command Syntax

```
range init_address [UPPER_RANGE]
no range init_address [UPPER_RANGE]
default range init_address [UPPER_RANGE]
```

Parameters

- *init_address* IP address of lower boundary of the address range (dotted decimal notation).
- ***UPPER_RANGE*** sets the upper boundary of the address range. Options include
  - <no parameter> upper boundary is equal to lower boundary: range consist of one address.
  - *range_address* IP address of upper boundary.

  All addresses must be multicast addresses (224.0.0.0 to 239.255.255.255).

Related Commands

- **ip igmp profile** places the switch in IGMP-profile configuration mode.

Example

- These commands enter IGMP profile configuration mode, configure the profile as a permit list, and define the permit address list of 232.1.1.0 to 232.1.1.255 and 233.1.1.10.

```
switch(config)#ip igmp profile list_1
switch(config-igmp-profile-list_1)#permit
switch(config-igmp-profile-list_1)#232.1.1.0 232.1.1.255
switch(config-igmp-profile-list_1)#233.1.1.10
switch(config-igmp-profile-list_1)#
```

show ip igmp profile

The **show ip igmp profile** command displays the contents of the specified IGMP profile. IGMP snooping filters use an IGMP profile to control the multicast groups that an interface can join.

Command Mode

EXEC

Command Syntax

show ip igmp snooping [***PROFILES***]

Parameters

- ***PROFILES*** IGMP profiles for which command displays contents. Options include:
  - <no parameter> displays all IGMP profiles.
  - *profile_name* displays specified profile.

Example

- This command displays the IGMP profiles configured on the switch.

```
switch>show ip igmp profile
IGMP Profile list_1
     permit
     range 229.1.24.0 229.1.25.255
IGMP Profile list_2
     range 234.1.1.0 234.1.255.255
switch>
```

show ip igmp snooping

The **show ip igmp snooping** command displays the Internet Group Management Protocol (IGMP) snooping configuration of a device.

Command Mode

EXEC

Command Syntax

show ip igmp snooping [***VLAN_ID***]

Parameters

- ***VLAN_ID*** specifies VLANs for which command displays information. Options include:
  - <no parameter> displays information for all VLANs.
  - **vlan** *v_num* displays information for specified VLAN.

Example

- This command displays the switch's IGMP snooping configuration.

```
switch>show ip igmp snooping
   Global IGMP Snooping configuration:
-----------------------------------------
IGMP snooping                   : Enabled
Robustness variable             : 2

Vlan 1 :
----------
IGMP snooping                   : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 20 :
----------
IGMP snooping                   : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 26 :
----------
IGMP snooping                   : Enabled
Multicast router learning mode : pim-dvmrp

Vlan 2028 :
----------
IGMP snooping                   : Enabled
Multicast router learning mode : pim-dvmrp

switch>
```

show ip igmp snooping counters

The **show ip igmp snooping counters** command displays the number of IGMP messages sent and received through each switch port. The display table sorts the messages by type.

Command Mode

EXEC

Command Syntax

show ip igmp snooping counters [*DATA_TYPE*][*DATA_LEVEL*]

Parameters

- *DATA_TYPE* Information displayed by the command. Options include:
  - — <no parameter> displays transmission counters.
  - — **errors** displays error counters.
- *DATA_LEVEL* specifies the type of information displayed. Options include:
  - — <no parameter> number of packets on physical ports.
  - — **detail** number of packets on physical ports.

Example

- This command displays the number of messages received on each port.

```
switch>show ip igmp snooping counters
                        Input                    |            Output
Port    Queries Reports  Leaves  Others  Errors|Queries Reports  Leaves  Others
-------------------------------------------------------------------------------
Cpu       15249  106599       4  269502       0   30242  102812     972    3625
Et1           0       0       0       0       0       0       0       0       0
Et2           0       6       1      26       0    5415       0       0     731
Et3           0   10905     222    1037       0   15246       0       0    1448
Et4           0   44475      21     288       0   15247       0       0    2199
Et5           0     355       0      39       0   15211       0       0    2446
Et6           0     475      13       0       0   15247       0       0    2487
Et7           0       0       0     151       0   15247       0       0    2336
Et8           0     578       6      75       0    2859       0       0     931
Et9           0       0       0      27       0   15247       0       0    2460
Et10          0   12523     345      54       0   15247       0       0    2433
Et11          0       0       0       0       0       0       0       0       0
Et12          0    4509      41      22       0   15247       0       0    2465
Et13          0     392      29     119       0   15247       0       0    2368
Et14          0      88       3       6       0   15247       0       0    2481
Et15          0   16779     556      72       0   15117       0       0      66
Et16          0    2484      13      66       0   15247       0       0    2421
Et17          0       0       0       0       0       0       0       0       0
Et18          0      20       6     160       0    3688       0       0     803
Et19          0    4110      17       0       0   15247       0       0    2487
Et20          0       0       0       0       0       0       0       0       0
Et21          0       0       0       0       0       0       0       0       0
Et22          0       0       0      52       0   15247       0       0    2435
Et23          0    5439     181     138       0   15247       0       0    2349
Et24          0    2251      21       4       0   15247       0       0    2483
Po1       45360  540670    8853  464900       0   15249  224751     618    2576
Po2           0  101399      58      17       0   15120       0       0    1121
Switch        0       0       0       0       0       0       0       0       0
```

show ip igmp snooping counters ethdev-pams

The **show ip igmp snooping counters** command displays the number of dropped IGMP packeets messages sent and received through each switch port at the kernel level.. The display table sorts the messages by type.

Command Mode

EXEC

Command Syntax

```
show ip igmp snooping counters ethdev-pams
```

Example

- This command displays the number of messages dropped at the kernel level.

```
switch>show ip igmp snooping counters ethdev-pams
  IntfName   rxErrors   txErrors    txDrops
       et9          1          0          0
      et18          1          0          0
     mlag9          1          0          0
     mlag8          1          0          0
      et17          1          0          0
       po1          1          0          0
       po2          1          0          0
      et15          1          0          0
       et6          1          0          0
    mlag10          1          0          0
      et16          1          0          0
     mlag7          1          0          0
      et11          1          0          0
     mlag5          1          0          0
     mlag4          1          0          0
       cpu          1          0          0
      et13          1          0          0
switch>
```

show ip igmp snooping groups

The **show ip igmp snooping groups** command displays IGMP snooping statistics. Available information includes the physical ports that send and receive information, the time when multicast data was originally and most recently heard on the ports, and the version number of the IGMP messages. Command provides options that restrict the output to specific VLANs, ports, and groups.

Command Mode

EXEC

Command Syntax

show ip igmp snooping groups [***VLAN_ID***] [***PORT_INT***] [***GROUPS***] [***DATA***]

Parameters

- ***VLAN_ID*** specifies VLAN for which command displays information. Options include:
  - <no parameter> displays information for all VLANs.
  - **vlan** *v_num* displays information for VLAN *v_num* (1 to 4094).
- ***PORT_INT*** specifies physical ports for which command displays information. Options include:
  - <no parameter> displays information for all physical ports.
  - **interface ethernet** *e_range*, where *e_range* is the number, range, or list of Ethernet ports.
  - **interface port-channel** *p_range*, where *p_range* is the number, range, or list of channel ports.
- ***GROUPS*** specifies the multicast groups. Options include:
  - <no parameter> all multicast groups on all specified ports.
  - *mgroup_address* multicast group specified by IPv4 address (dotted decimal notation).
  - **dynamic** multicast groups learned through IGMP.
  - **user** multicast groups manually added.
- ***DATA*** specifies the type of information displayed. Options include:
  - <no parameter> VLAN number and port-list for each group.
  - **detail** port-specific information for each group, including transmission times and expiration.

Example

- This command displays the port lists for all multicast groups.

```
switch>show ip igmp snooping groups
Vlan  Group           Type     Version             Port-List
------------------------------------------------------------------------------
1     239.255.255.250 -        -                   Po1, Po2
26    239.255.255.250 -        -                   Cpu, Et3, Et4, Et10, Et23,
                                                   Et27
switch>
```

- This command displays detailed port information of all multicast groups.

```
switch>show ip igmp snooping groups detail
Vlan Group           IP              First     Last    Expire    Ver Filter Port
                                     Heard     Heard                 Mode
-------------------------------------------------------------------------------
1    239.255.255.250 172.17.3.73     2536:15   0:47    3:33      v2  0      Po2
1    239.255.255.250 172.17.0.37     31532:48  0:18    1:27      -   -      Po1
26   239.255.255.250 172.17.26.189   5:07      0:52    3:28      v2  0      Et3
26   239.255.255.250 172.17.26.182   17:34     3:02    1:18      v2  0      Et3
26   239.255.255.250 172.17.26.245   1046:47   0:57    3:23      v2  0      Et4
26   239.255.255.250 172.17.26.184   27:41     0:53    3:27      v2  0      Et10
26   239.255.255.250 172.17.26.161   9:16      0:56    3:24      v2  0      Et23
26   239.255.255.250 172.17.26.62    90:24     0:50    3:30      v2  0      Et27
26   239.255.255.250 172.17.26.1     31532:52  0:04    1:41      -   -      Cpu
switch>
```

- This command displays the port lists for all dynamic multicast groups.

```
switch>show ip igmp snooping groups dynamic
Vlan  Group            Type     Version             Port-List
-------------------------------------------------------------------------------
1     239.255.255.250  -        -                   Po1, Po2
26    239.255.255.250  -        -                   Cpu, Et3, Et4, Et10, Et23,
                                                    Et27, Et34
switch>
```

- This command displays the detailed port information for all dynamic multicast groups.

```
switch>show ip igmp snooping groups dynamic detail
Vlan Group           IP              First     Last    Expire    Ver Filter Port
                                     Heard     Heard                 Mode
-------------------------------------------------------------------------------
1    239.255.255.250 172.17.3.73     2539:16   1:37    2:43      v2  0      Po2
1    239.255.255.250 172.17.0.37     31535:49  0:19    1:26      -   -      Po1
26   239.255.255.250 172.17.26.189   8:08      3:53    0:27      v2  0      Et3
26   239.255.255.250 172.17.26.182   20:35     1:49    2:31      v2  0      Et3
26   239.255.255.250 172.17.26.245   1049:48   1:46    2:34      v2  0      Et4
26   239.255.255.250 172.17.26.184   30:42     1:44    2:36      v2  0      Et10
26   239.255.255.250 172.17.26.161   12:17     3:57    0:23      v2  0      Et23
26   239.255.255.250 172.17.26.143   1:53      1:53    2:27      v2  0      Et23
26   239.255.255.250 172.17.26.62    93:25     1:48    2:32      v2  0      Et27
26   239.255.255.250 172.17.26.164   0:32      0:31    3:49      v2  0      Et34
26   239.255.255.250 172.17.26.1     31535:53  0:05    1:40      -   -      Cpu
switch>
```

- This command displays the port lists for all static (user configured) multicast groups.

```
switch>show ip igmp snooping groups user
Vlan  Group            Type     Version             Port-List
-------------------------------------------------------------------------------
1     239.255.255.250  -        -                   Po1, Po2
26    239.255.255.250  -        -                   Cpu, Et3, Et4, Et10, Et23,
                                                    Et27, Et34
switch>
```

- This command displays detailed port information for all user configured (static) multicast groups.

```
switch>show ip igmp snooping groups user detail
Vlan Group           IP              First     Last    Expire    Ver Filter Port
                                     Heard     Heard                 Mode
-------------------------------------------------------------------------------
1    239.255.255.250 172.17.3.73     2539:50   0:06    4:14      v2  0      Po2
1    239.255.255.250 172.17.0.37     31536:23  0:23    1:22      -   -      Po1
26   239.255.255.250 172.17.26.182   21:09     0:21    3:59      v2  0      Et3
26   239.255.255.250 172.17.26.245   1050:22   0:17    4:03      v2  0      Et4
26   239.255.255.250 172.17.26.184   31:16     0:17    4:03      v2  0      Et10
26   239.255.255.250 172.17.26.161   12:51     0:17    4:03      v2  0      Et23
26   239.255.255.250 172.17.26.143   2:27      2:27    1:53      v2  0      Et23
26   239.255.255.250 172.17.26.62    93:59     0:22    3:58      v2  0      Et27
26   239.255.255.250 172.17.26.164   1:06      0:21    3:59      v2  0      Et34
26   239.255.255.250 172.17.26.1     31536:27  0:09    1:36      -   -      Cpu
switch>
```

- This command displays detailed port information for multicast group 239.255.255.253 on VLAN 10.

```
switch>show ip igmp snooping groups vlan 10 239.255.255.253 detail
Vlan Group           IP              First     Last    Expire    Ver Filter Port
                                     Heard     Heard                 Mode
-------------------------------------------------------------------------------
10   239.255.255.253 10.255.255.246  7177:16   0:08    2:07      v2  0      Po7
10   239.255.255.253 10.255.255.247  7177:20   0:03    2:12      v2  0      Po7
10   239.255.255.253 10.255.255.248  7177:16   0:06    2:09      v2  0      Po7
10   239.255.255.253 10.255.255.254  7177:56   0:07    1:38      -   -      Cpu
switch>
```

show ip igmp snooping groups count

The **show ip igmp snooping groups count** command displays the number of multicast groups on the switch. Command provides options to only include specific VLANs and ports.

Command Mode

EXEC

Command Syntax

show ip igmp snooping groups [***VLAN_ID***][***PORT_INT***] **count** [***DATA***]

Parameters

- ***VLAN_ID*** specifies VLAN for which command displays information. Options include:
  - <no parameter> all VLANs.
  - **vlan** *v_num* specified VLAN.
- ***PORT_INT*** specifies physical ports for which command displays information. Options include:
  - <no parameter> all physical ports.
  - **interface ethernet** *e_range* specified Ethernet ports.
  - **interface port-channel** *p_range* specified port channels.

  Valid *e_range* and *p_range* formats include number, number range, or comma-delimited list of numbers and ranges.
- ***DATA*** specifies the type of information displayed. Options include:
  - <no parameter> number of multicast group on specified VLAN and ports.
  - **detail** number of multicast group on specified VLAN and ports.

Example

- This command displays the number of multicast groups on the switch.

```
switch>show ip igmp snooping groups count
Total number of multicast groups: 2
switch>
```

show ip igmp snooping mrouter

The **show ip igmp snooping mrouter** command displays information on dynamically learned and manually configured multicast router ports. Command provides options to include only specific VLANs.

Command Mode

EXEC

Command Syntax

show ip igmp snooping mrouter [***VLAN_ID***] [***DATA***]

Parameters

- ***VLAN_ID*** specifies VLAN for which command displays information. Options include:
  - <no parameter> all VLANs.
  - **vlan** *v_num* specified VLAN.
- ***DATA*** specifies the type of information displayed. Options include:
  - <no parameter> displays VLAN number and port-list for each group.
  - **detail** displays port-specific data for each group; includes transmission times and expiration.

Examples

- This command displays port information of each multicast router on all VLANs.

```
switch>show ip igmp snooping mrouter
Vlan    Interface-ports
-------------------------------------------------------------
1       Po1(dynamic)
20      Po1(dynamic)
26      Cpu(dynamic)
2028    Cpu(dynamic), Po1(dynamic)
switch>
```

- This command displays multicast router information for each port.

```
switch>show ip igmp snooping mrouter detail
Vlan   Intf      Address           FirstHeard LastHeard  Expires    Type
-------------------------------------------------------------------------------
1      Po1       172.17.0.37       31549:12   0:12       1:33       pim
20     Po1       172.17.20.1       7066:51    0:19       1:26       pim
26     Cpu       172.17.26.1       31549:16   0:28       1:17       pim
2028   Po1       172.17.255.29     31549:10   0:18       1:27       pim
2028   Cpu       172.17.255.30     31549:14   0:28       1:17       pim
switch>
```

show ip igmp snooping querier

The **show ip igmp snooping querier** command displays snooping querier configuration and status information. Command provides options to only include specific VLANs.

Command Mode

EXEC

Command Syntax

show ip igmp snooping querier [***STATUS***] [***VLAN_ID***] [***DATA***]

Parameters

- ***STATUS*** specifies the type of information displayed. Options include:
  - — <no parameter> querier IP address, port, and IGMP version.
  - — **status** querier configuration parameters.
- ***VLAN_ID*** specifies VLANs for which command displays information. Options include:
  - — <no parameter> all VLANs.
  - — **vlan** *v_num* specified VLAN.
- ***DATA*** specifies the type of information displayed. Options include:
  - — <no parameter> displays VLAN number and port-list for each group.
  - — **detail** displays port-specific data for each group; includes transmission times and expiration.

Examples

- This command displays the querier IP address, version, and port servicing each VLAN.

```
switch>show ip igmp snooping querier
Vlan  IP Address       Version  Port
---------------------------------------
1     172.17.0.37      v2       Po1
20    172.17.20.1      v2       Po1
26    172.17.26.1      v2       Cpu
2028  172.17.255.29    v2       Po1
switch>
```

- This command displays the querier configuration parameters for each VLAN.

```
switch>show ip igmp snooping querier status
   Global IGMP Querier status
-----------------------------------
admin state              : Enabled
source IP address        : 0.0.0.0
query-interval (sec)     : 125.0
max-response-time (sec)  : 10.0
querier timeout (sec)    : 130.0

Vlan Admin    IP             Query    Response Querier Operational
     State                   Interval Time     Timeout State
-----------------------------------------------------------------------
1    Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
4    Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
20   Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
22   Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
28   Enabled  0.0.0.0         125.0    10.0     130.0   Non-Querier
```

23.9 PIM Commands

This section contains descriptions of the CLI commands that this chapter references.

PIM Configuration Commands (Global)

- ip pim anycast-rp . . . . . Page 1076
- ip pim bsr-candidate . . . . . Page 1079
- ip pim bsr-border . . . . . Page 1078
- ip pim log-neighbor-changes . . . . . Page 1083
- ip pim register-source . . . . . Page 1086
- ip pim rp-address . . . . . Page 1087
- ip pim rp-candidate . . . . . Page 1088
- ip pim sparse-mode sg-expiry-timer . . . . . Page 1091
- ip pim spt-threshold. . . . . Page 1092
- ip pim ssm range . . . . . Page 1093

PIM Configuration Commands (Interface)

- ip pim border-router . . . . . Page 1077
- ip pim bsr-border . . . . . Page 1078
- ip pim dr-priority . . . . . Page 1081
- ip pim join-prune-interval . . . . . Page 1082
- ip pim neighbor-filter. . . . . Page 1084
- ip pim query-interval. . . . . Page 1085
- ip pim sparse-mode . . . . . Page 1090

PIM Display Commands

- show ip pim bsr . . . . . Page 1094
- show ip pim config-sanity. . . . . Page 1095
- show ip pim interface . . . . . Page 1096
- show ip pim neighbor . . . . . Page 1097
- show ip pim protocol. . . . . Page 1098
- show ip pim register-source . . . . . Page 1099
- show ip pim rp . . . . . Page 1100
- show ip pim rp-candidate. . . . . Page 1101
- show ip pim rp-hash . . . . . Page 1102
- show ip pim upstream joins . . . . . Page 1103

ip pim anycast-rp

The **ip pim anycast-rp** command configures the switch as a member of an anycast-RP set and establishes a communication link with another member of the set.

PIM Anycast-RP defines a single RP address that is configured on multiple routers. An anycast-RP set consists of the routers configured with the same anycast-RP address. Anycast-RP provides redundancy protection and load balancing. The anycast-RP set supports all multicast groups.

PIM register messages are unicast to the RP by designated routers (DRs) that are directly connected to multicast sources. The switch sends these messages and join-prune messages to the anycast-RP set member specified in the anycast-RP command. In a typical configuration, one command is required for each member of the anycast-RP set.

The PIM register message has the following functions:

- Notify the RP that a source is actively sending to a multicast group.
- Deliver multicast packets sent by the source to the RP for delivery down the shared tree.

The DR continues sending PIM register messages to the RP until it receives a Register-Stop message from the RP. The RP sends a Register-Stop message in either of the following cases:

- The RP has no receivers for the multicast group being transmitted.
- The RP has joined the SPT to the source but has not started receiving traffic from the source.

The **no ip pim anycast-rp** and **default ip pim anycast-rp** commands remove the corresponding **ip pim anycast-rp** commands from ***running-config***. When the **no** and **default** commands do not include a peer address, all commands for the specified rp address are removed.

Command Mode

Global Configuration

Command Syntax

```
ip pim anycast-rp rp_addr peer_addr [REGISTER]
no ip pim anycast-rp rp_addr [peer_addr]
default ip pim anycast-rp rp_addr [peer_addr]
```

Parameters

- *rp_addr* Rendezvous point IP address (dotted decimal notation).
- *peer_addr* IP address of an anycast-RP set member (dotted decimal notation).
- ***REGISTER*** Number of unacknowledged register messages the switch sends to the peer router.
  - — <No parameter> register count is set to default value of 10.
  - — **register-count** *r_num* where *r_num* is an integer that ranges from 1 to 4294967295.
  - — **register-count infinity**

Example

- These commands configure a switch (IP address 10.1.1.14) into an anycast-RP set with an RP address of 172.17.255.29. The anycast-RP set contains three other routers, located at 10.1.2.14, 10.1.3.14, and 10.1.4.14. It sets the number of unacknowledged register messages it sends to each router at 15.

```
switch(config)#ip pim anycast-rp 172.17.255.29 10.1.1.14 register-count 15
switch(config)#ip pim anycast-rp 172.17.255.29 10.1.2.14 register-count 15
switch(config)#ip pim anycast-rp 172.17.255.29 10.1.3.14 register-count 15
switch(config)#ip pim anycast-rp 172.17.255.29 10.1.4.14 register-count 15
```

ip pim border-router

The **ip pim border-router** command configures the configuration mode interface as a PIM multicast border router (MBR). A PIM MBR interface allows multicast traffic from sources that are outside of the PIM domain. An interface's configuration as an MBR is independent to it PIM sparse-mode setting. This command does not control the transmission or reception of PIM protocol packets by the interface.

Sources learned through an MBR interface are treated as local sources (directly connected to the switch). The border-bit is set in all PIM register messages sent for these sources.

The **no ip pim border-router** and **default ip pim border-router** commands removes the PIM MBR configuration for the configuration mode interface by removing the corresponding **ip pim border-router** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim border-router
no ip pim border-router
default ip pim border-router
```

Example

- These commands configure VLAN interface 200 as a PIM MBR, then displays its status through a show IP PIM interface command.

```
switch(config)#interface vlan 200
switch(config-if-VL200)#ip address 171.44.2.1/24
switch(config-if-VL200)#ip pim border-router
switch(config-if-VL200)#ip pim sparse-mode
switch(config-if-VL200)#show active
interface Vlan200
   ip address 171.44.2.1/24
   ip pim sparse-mode
   ip pim border-router

switch(config-if-VL200)#exit
switch(config)#show ip pim interface
Address         Interface    Mode       Neighbor   Hello DR         DR Address
                                        Count      Intvl Pri
171.44.2.1      Vlan200      sparse/mbr 0          30    1          171.44.2.1
```

ip pim bsr-border

The **ip pim bsr-border** command prevents the switch from sending bootstrap router messages (BSMs) over the configuration mode interface. By default, BSMs are transmitted over all PIM-SM enabled interfaces.

The **no ip pim bsr-border** and **default ip pim bsr-border** commands restore the transmission of BSMs over the configuration mode interface by removing the corresponding **ip pim bsr-border** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim bsr-border
no ip pim bsr-border
default ip pim bsr-border
```

Example

- This command prevents the switch from sending BSMs from VLAN interface 10.

```
switch(config)#interface vlan 10
switch(config-if-Vl10)#ip pim bsr-border
switch(config-if-Vl10)#
```

ip pim bsr-candidate

The **ip pim bsr-candidate** command configures the switch as a candidate BSR router (C-BSR). A BSR is a PIM router within the PIM domain through which dynamic RP selection is implemented. The BSR selects RPs from a list of candidate RPs and exchange bootstrap messages (BSM) with all routers in the domain. The BSR is elected from one of the C-BSRs through an exchange of BSMs.

A subset of PIM routers within the domain are configured as candidate Bootstrap routers (C-BSRs). Through the exchange of Bootstrap messages (BSMs), the C-BSRs elect the BSR, which then uses BSMs to inform all domain routers of its status.

Command parameters specify the switch's BSR address, the interval between BSM transmissions, and the priority assigned to the switch when electing a BSR.

Entering an **ip pim bsr-candidate** command replaces any previously configured **bsr-candidate** command. If the new command does not specify a priority or interval, the previously configured values persist in ***running-config***.

The **no ip pim bsr-candidate** and **default ip pim bsr-candidate** commands remove the corresponding **ip pim bsr-candidate** commands from ***running-config***. The **no** and **default** commands restore the priority and interval parameters to their default values.

Command Mode

Global Configuration

Command Syntax

```
ip pim bsr-candidate INTERFACE [PRIORITY_NUM] [INTERVAL_PERIOD]
no ip pim bsr-candidate [priority] [interval]
default ip pim bsr-candidate [priority] [interval]
```

Parameters

- ***INTERFACE*** Switch uses IP address of specified interface as its BSR address. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
- ***PRIORITY_NUM*** BSR election priority rating. Larger numbers denote higher priority. Default value is 64.
  - <no parameter> priority remains unchanged from previous setting.
  - **priority** *<0 - 255>* priority rating.
- ***INTERVAL_NUM*** Period between the transmission of BSMs (seconds). Default value is 60.
  - <no parameter> interval remains unchanged from previous setting..
  - **interval** *<10 - 536870906>* transmission interval.

Example

- These commands configures the switch as a BSR candidate, using the IP address assigned to VLAN interface 24 as its BSR address. The BSM transmission interval is set to 30 seconds and the priority is set to 192.

```
switch(config)#ip pim bsr-candidate vlan 24 priority 192 interval 30
switch(config)#
```

ip pim bsr-holdtime

The **ip pim bsr-holdtime** command specifies the value the switch inserts in the ***holdtime*** parameter field in Bootstrap messages (BSM) that it sends. The BSR holdtime defines the timeout period that an elected BSR remains valid after the receipt of a BSM and is also used in dynamic RP configuration. BSR holdtime is designated by the BSR router and communicated to other routers through BSMs.

The **no ip pim bsr-holdtime** and **default ip pim bsr-holdtime** commands restore the default holdtime parameter field insertion value of 130 seconds by removing the **ip pim dr-priority** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip pim bsr-holdtime period
no ip pim bsr-holdtime
default ip pim bsr-holdtime
```

Parameters

- *period* BSR holdtime (seconds). Values range from 12 to 1073741823 (1.073 billion). Default is 130.

Example

- This command specifies 75 seconds as the value that the switch inserts into BSM holdtime fields.

```
switch(config)#ip pim bsr-holdtime 75
switch(config)#
```

ip pim dr-priority

PIM uses these criteria for electing designated routers (DR):

- If one router does not advertise a dr-priority value, the router with the highest IP address becomes the Designated Router.
- If all router advertise a dr-priority value, the router with the highest dr-priority value becomes the Designated Router.

The **ip pim dr-priority** command sets the dr-priority value that the configuration mode interface advertises. By default, the interface does not advertise a dr-priority value.

The **no ip pim dr-priority** and **default ip pim dr-priority** commands force the use of IP addresses to elect the designated router by removing the corresponding **ip pim dr-priority** statement from *running-config*.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim dr-priority level
no ip pim dr-priority [level]
default ip pim dr-priority [level]
```

Parameters

- *level* DR selection priority rating. Values range from 0 to 1000000 (1 million).

Examples

- This command configures the dr-priority value of 15 on VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip pim dr-priority 15
switch(config-if-Vl4)#
```

- This command removes the **ip-pim dr-priority** statement from *running-config*.

```
switch(config-if-Vl4)#no ip pim dr-priority
switch(config-if-Vl4)#
```

ip pim join-prune-interval

The **ip pim join-prune-interval** command specifies the period between join/prune messages that the configuration mode interface originates and sends to the upstream RPF neighbor.

The **no ip pim join-prune-interval** and **default ip pim join-prune-interval** commands restore the default join/prune interval of 60 seconds for the configuration mode interface by removing the corresponding **ip pim join-prune-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim join-prune-interval period
no ip pim join-prune-interval [period]
default ip pim join-prune-interval [period]
```

Parameters

- *period* join/prune interval (seconds). Values range from 1 to 1000000 (1 million). Default is 60.

Example

- This command configures 75-second intervals between join/prune messages originating from VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip pim join-prune-interval 75
switch(config-if-Vl4)#
```

ip pim log-neighbor-changes

The **ip pim log-neighbor-changes** command configures the switch to generate a log message when a neighbor entry is added or removed from the PIM Neighbor table. This function is enabled by default.

The **no ip pim log-neighbor-changes** command disables log message generation based on changes to the PIM Neighbor table; this command is stored to ***running-config***. The **ip pim log-neighbor-changes** and **default ip pim log-neighbor-changes** commands restore the default setting of generating log messages by deleting the **no ip pim log-neighbor-changes** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip pim log-neighbor-changes
no ip pim log-neighbor-changes
default ip pim log-neighbor-changes
```

Examples

- This command configures the switch to stop generating log messages based on PIM Neighbor table changes.

```
switch(config)#no ip pim log-neighbor-changes
switch(config)#
```

- This command configures the switch to generate log messages when a neighbor entry is added or removed from the PIM Neighbor table.

```
switch(config)#ip pim log-neighbor-changes
switch(config)#
```

ip pim neighbor-filter

The **ip pim neighbor-filter** command configures the configuration mode interface to filter PIM control packets on the basis of neighbor addresses listed in a specified standard access list.

The **no ip pim neighbor-filter** and **default ip pim neighbor-filter** commands disable the configuration mode interface from filtering PIM control packets by removing the corresponding **ip pim neighbor-filter** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim neighbor-filter access_list
no ip pim neighbor-filter
default ip pim neighbor-filter
```

Parameters

- *access_list* name of the standard IP access list.

Example

- This command configures the IP access list named filter_1 to filter neighbor PIM control messages for VLAN 4.

```
switch(config)#ip access-list standard filter_1
switch(config-std-acl-filter_1)#permit 171.13.24.9/24
switch(config-std-acl-filter_1)#exit
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip pim neighbor-filter filter_1
switch(config-if-Vl4)#
```

ip pim query-interval

The **ip pim query-interval** command specifies the transmission interval between PIM hello messages originating from the configuration mode interface.

The **no ip pim query-interval** and **default ip pim query-interval** commands restore the default query interval of 30 seconds for the configuratiom mode interface by removing the corresponding **ip pim query-interval** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim query-interval period
no ip pim query-interval [period]
default ip pim query-interval [period]
```

Parameters

- *period* query interval (seconds). Values range from 1 to 1000000 (1 million). Default is 30.

Example

- This command configures 45 second intervals between hello messages originating from VLAN interface 4.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip pim query-interval 45
switch(config-if-Vl4)#
```

ip pim register-source

The **ip pim register-source** command programs the switch to fill the source field in all outbound PIM SM register packets with the IP address of the specified interface. By default, the source field is filled with the IP address from the interface associated with the best route to the RP.

The **no ip pim register-source** and **default ip pim register-source** commands restore the default method of filling the register packet source field by deleting the **ip pim register-source** statement from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
ip pim register-source INT_NAME
no ip pim register-source
default ip pim register-source
```

Parameters

- *INT_NAME* Interface type and number. Values include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port channel interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command programs the switch to fill the source field of outbound PIM SM register packets with the IP address of loopback interface 2.

```
switch(config)#ip pim register-source loopback 2
switch(config)#
```

ip pim rp-address

The **ip pim rp-address** command configures the address of a Protocol Independent Multicast (PIM) static rendezvous point (RP) for a specified multicast subnet. If the command does not specify a subnet, the static RP maps to all multicast groups (224/4).

Multicast groups use RPs to connect sources and receivers. A PIM domain requires that all routers have consistently configured RP addresses.

The switch uses multiple **ip pim rp-address** commands to configure multiple RPs or to assign multiple subnets to an RP. When the address of a multicast group falls within multicast subnets configured by multiple **ip pim rp-address** commands, the switch selects the groups's RP address by comparing the commands' multicast subnet size.

- Different size subnets: group uses command with the largest subnet.
- Same size subnets: group uses command as determined by hash algorithm.

The **no ip pim rp-address** and **default ip pim rp-address** commands remove the corresponding **ip pim rp-address** command from ***running-config***. If the command does not include a multicast subnet parameter, it removes all statements with the specified RP address from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip pim rp-address rp_addr [MULTICAST_SUBNET] [BSR_OVERRIDE]
no ip pim rp-address rp_addr [MULTICAST_SUBNET]
default ip pim rp-address rp_addr [MULTICAST_SUBNET]
```

Parameters

- *rp_addr* Rendezvous point IP address (dotted decimal notation).
- ***MULTICAST_SUBNET*** Multicast IP address space (CIDR or address-mask).
  - <no parameter> Default multicast group IP address of 224/4.
  - *gp_addr* Multicast group IP address (CIDR or address-mask).
  - **access-list** *acl_name* Standard access control list that specifies the multicast group address.
- ***BSR_OVERRIDE*** Configures priority relative to dynamic RPs selected by BSR.
  - <no parameter> Dynamic RPs have priority over specified RP.
  - **override** RP has priority over dynamic RPs.

Example

- This command configures 172.17.255.29 as a static RP to all multicast groups.

```
switch(config)#ip pim rp-address 172.17.255.29
switch(config)#
```

ip pim rp-candidate

The **ip pim rp-candidate** command configures the switch as a candidate rendezvous point (C-RP). The BSR selects a multicast group's dynamic RP set from the list of C-RPs in the PIM domain. The command specifies the interface (used to derive the RP address), C-RP advertisement interval, and priority rating. The BSR select the RP set by comparing C-RP priority ratings. The C-RP advertisement interval specifies the period between successive C-RP advertisement message transmissions to the BSR.

Running-config supports multiple multicast groups through multiple **ip pim rp-candidate** statements:

- All commands must specify the same interface. Issuing a command with an interface that differs from existing commands removes all existing commands from ***running-config***.
- ***Running-config*** stores the ***interval*** setting in a separate statement that applies to all **rp-candidate** statements. When a command specifies an interval that differs from the previously configured value, the new value replaces the old value and applies to all configured **rp-candidate** statements.

The **no ip pim rp-candidate** and **default ip pim rp-candidate** commands remove the **ip pim rp-candidate** from ***running-config*** command for the specified multicast group . When these commands do not specify a multicast group, all **rp-candidate** statements are removed from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip pim rp-candidate INTERFACE [GROUP_ADDR][PRIORITY_NUM][INTERVAL_PERIOD]
no ip pim rp-candidate [INTERFACE][GROUP_ADDR][priority][interval]
default ip pim rp-candidate [INTERFACE][GROUP_ADDR] [priority] [interval]
```

Parameters

- ***INTERFACE*** Switch uses IP address of specified interface as its C-RP address. Options include:
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.
- ***GROUP_ADDR*** the multicast group subnet address. Options include:
  - <no parameter> register count is set to default value of 10.
  - *net_addr* multicast IPv4 subnet address (CIDR or address mask).
  - **access-list** *acl_name* standard access control list that specifies the multicast group address.
- ***PRIORITY_NUM*** RP selection priority rating. Smaller numbers denote higher priority.
  - <no parameter> priority rating is set to the default value of 0.
  - **priority** *<0 - 255>* priority rating.
- ***INTERVAL_NUM*** Period between succesive RP-advertisement message transmissions (seconds). Value also applies to previously configured rp-candidate statements.
  - <no parameter> interval remains unchanged from previous setting.
  - **interval** *<10 - 16383>* transmission interval.

Example

- This command configures a switch as a candidate RP for the multicast group 235.1.1.0/24 with a priority of 48 and a RP advertisement interval of 45 seconds. The switch advertises the IP address assigned to VLAN 24 as its RP address.

```
switch(config)#ip pim rp-candidate vlan 24 235.1.1.0/24 priority 48 interval 45
switch(config)#
```

ip pim sparse-mode

The **ip pim sparse-mode** command enables PIM and IGMP (router mode) on the configuration mode interface.

The **no ip pim sparse-mode**, **no ip pim, default ip pim sparse-mode**, and **default ip pim** commands restore the default PIM and IGMP (router mode) settings of ***disabled*** on the configuration mode interface by removing the corresponding the **ip pim sparse-mode** statement from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration
Interface-VLAN Configuration

Command Syntax

```
ip pim sparse-mode
no ip pim
no ip pim sparse-mode
default ip pim
default ip pim sparse-mode
```

Example

- This command enables PIM sparse mode on VLAN 4 interface.

```
switch(config)#interface vlan 4
switch(config-if-Vl4)#ip pim sparse-mode
switch(config-if-Vl4)#
```

ip pim sparse-mode sg-expiry-timer

The **ip pim sparse-mode sg-expiry-timer** command adjusts the (S, G) expiry timer interval for PIM-SM (S, G) multicast routes (mroutes). This command locks the shortest-path tree (SPT) for intermittent PIM-SM sources. The command does not apply to (*, G) mroutes.

When a source stops sending traffic to a multicast group, the corresponding (S, G) mroute is removed upon timer expiry. When the source resumes sending traffic to the group, the (S, G) entry is rebuilt. Before the (S, G) entry is rebuilt, traffic is forwarded on the (*, G) forwarding entry. Packets may be dropped before the (S, G) entry is completely built. The **ip pim sparse-mode sg-expiry-timer** command maintains the (S, G) entry, avoiding its removal and preventing packet loss.

The **no ip pim sparse-mode sg-expiry-timer** and **default ip pim sparse-mode sg-expiry-timer** commands restore the default setting of 210 seconds by deleting the **ip pim sparse-mode sg-expiry-timer** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip pim sparse-mode sg-expiry-timer period
no ip pim sparse-mode sg-expiry-timer
default ip pim sparse-mode sg-expiry-timer
```

Parameters

- *period* expiry timer interval (seconds). Values range from 120 (two minutes) to 65535 (18 hours, 12 minutes, 15 seconds). Default is 210 (three minutes, 30 seconds).

Example

- This command configures 2 minutes 30 seconds as the (S,G) expiry timer interval.

```
switch(config)#ip pim sparse-mode sg-expiry-timer 150
switch(config)#
```

ip pim spt-threshold

The **ip pim spt-threshold** command determines if the switch, acting as a Protocol Independent Multicast (PIM) leaf router, joins the shortest path source tree.

- When running-config does not list this command, the switch joins the shortest path tree (SPT) immediately after receiving the first PIM packet from a new source. The switch joins the SPT by sending PIM join message toward the source.
- When running-config lists this command with a value of infinity, the switch never joins the SPT.

The **no ip pim spt-threshold** and **default ip pim spt-threshold** commands restore the default value of 0 by removing the **ip pim spt-threshold infinity** command from running-config.

Command Mode

Global Configuration

Command Syntax

```
ip pim spt-threshold JOIN
no ip pim spt-threshold
default ip pim spt-threshold
```

Parameters

- *JOIN* specifies switch's inclusion into the shortest path tree. Options include:
  - **0** The switch immediately joins the SPT. This is the default value.
  - **infinity** The switch never joins the SPT.

Examples

- This command configures the switch to never join the SPT.

```
switch(config)#ip pim spt-threshold infinity
switch(config)#
```

- These equivalent commands restore the default value by removing the **ip pim spt-threshold** statement from running-config.

```
switch(config)#ip pim spt-threshold 0
switch(config)#

switch(config)#no ip pim spt-threshold
switch(config)#
```

ip pim ssm range

The **ip pim ssm range** command defines the source specific multicast SSM range of IP multicast addresses.

SSM is a multicast packet delivery method where only packets originating from a specific source address requested by a receiver are routed to that receiver. SSM explicitly excludes the use of (*,G) join for all multicast groups. Source-specific multicast differs from any-source multicast (ASM), where a receiver expresses interest in traffic to a multicast address, then receives traffic from all multicast sources sending to that address.

The **no ip pim ssm range** and **default ip pim ssm range** commands remove the SSM IP multicast address range by deleting the **ip pim ssm range** statement from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
ip pim ssm range [ACCESS_RANGE]
no ip pim ssm range
default ip pim ssm range
```

Parameters

- *ACCESS_RANGE* specifies the SSM IP multicast address range. Options include:
  - *acl_name* sets the SSM range to address set specifed by the specified standard ACL.
  - **standard** sets the SSM range to 232/8.

Examples

- This command configures the SSM address range to 232/8.

```
switch(config)#ip pim ssm range standard
switch(config)#
```

- These commands configure the SSM address range to those permitted by the LIST_1 standard ACL. The ACL permits the subnet address range 233.0.0.0/24.

```
switch(config)#ip access-list standard LIST_1
switch(config-std-acl-LIST_1)#permit 233.0.0.0/24
switch(config-std-acl-LIST_1)#exit
switch(config)#ip pim ssm range LIST_1
switch(config)#
```

show ip pim bsr

The **show ip pim bsr** command displays the switch's bootstrap router (BSR) information.

Command Mode

EXEC

Command Syntax

show ip pim bsr

Example

- This command configures the switch's BSR information.

```
switch>show ip pim bsr
PIMv2 Bootstrap information
This system is the Bootstrap Router (BSR)
  BSR address: 10.1.1.1
  Uptime:      00:14:42, BSR Priority: 0, Hash mask length: 30
  Next bootstrap message in 00:00:05
```

show ip pim config-sanity

The **show ip pim config-sanity** command displays diagnostic information about a PIM configuration.

Command Mode

EXEC

Command Syntax

```
show ip pim config-sanity
```

Example

- This command displays PIM configuration diagnostic information.

```
switch>show ip pim config-sanity
DISCLAIMER: Below are only hints of potential PIM misconfiguration.
They do not necessary imply that there is a real problem.

The interfaces with PIM which are down: Vl4

switch>
```

show ip pim interface

The **show ip pim interface** command displays information about interfaces configured for PIM.

Command Mode

EXEC

Command Syntax

show ip pim interface [***INT_NAME***] [***INFO_LEVEL***]

Parameters

- *INT_NAME* Interface type and number. Values include
  - — <no parameter> displays information for all interfaces.
  - — **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - — **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - — **vlan** *v_num* VLAN interface specified by *v_num*.
- *INFO_LEVEL* specifies level of information detail provided by the command.
  - — <no parameter> table of basic configuration information.
  - — **detail** list of complete configuration information.

Examples

- This command displays information about all interfaces on which PIM is enabled.

```
switch>show ip pim interface
Address         Interface    Mode      Neighbor   Hello DR         DR Address
                                       Count      Intvl Pri
172.17.26.1     Vlan26       sparse    0          30    1          172.17.26.1
172.17.255.30   Vlan2028     sparse    1          30    1          172.17.255.30

switch>
```

- This command displays detailed PIM information for VLAN 26 interface.

```
switch>show ip pim interface vlan 26 detail
Interface address is 172.17.26.1
Vif number is 1
PIM: enabled
  PIM version: 2, mode: sparse
  PIM DR: 172.17.26.1 (this system)
  PIM DR Priority: 1
  PIM neighbor count: 0
  PIM Hello Interval: 30 seconds
  PIM Hello Priority: 1
  PIM Hello Lan Delay: 500 milliseconds
  PIM Hello Override Interval: 2500 milliseconds
  PIM Hello Lan Prune Delay in use
  PIM Hello Generation ID: 0x4a05aa0
  PIM Hello Generation ID is not required
  PIM Triggered Hello Delay: 5 seconds
  PIM Join-Prune Interval: 60 seconds
  PIM State-Refresh processing: disabled
  PIM State-Refresh Interval: unknown seconds
  PIM Graft Retry Interval: unknown seconds
  PIM domain border: disabled

switch>
```

show ip pim neighbor

The **show ip pim neighbor** command displays information about Protocol Independent Multicast (PIM) neighbors discovered by hello messages.

Command Mode

EXEC

Command Syntax

show ip pim neighbor [***INT_NAME***]

Parameters

- ***INT_NAME*** Interface type and number. Values include
  - — <no parameter> displays information for all interfaces.
  - — **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - — **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - — **vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command displays information about neighbor PIM routers.

```
switch>show ip pim neighbor
PIM Neighbor Table
Neighbor Address   Interface       Uptime     Expires    Mode
172.17.255.29      Vlan2028        21d22h     00:01:31   sparse

switch>
```

show ip pim protocol

The **show ip pim protocol** command displays statistics about Protocol Independent Multicast (PIM) control messages sent and received by the switch.

Command Mode

EXEC

Command Syntax

show ip pim protocol

Example

- This command displays statistics about inbound and outbound PIM control messages.

```
switch>show ip pim protocol
PIM Control Counters
                          Received          Sent       Invalid
Assert                           0            37             0
Bootstrap Router                 0             0             0
CRP Advertisement                0             0             0
Graft                            0             0             0
Graft Ack                        0             0             0
Hello                        63168        126355             0
J/P                         275714        143958             0
Join                             0             0             0
Prune                            0             0             0
Register                         0         13643             0
Register Stop                11839             0             0
State Refresh                    0             0             0

switch>
```

show ip pim register-source

The **show ip pim register-source** command displays the name of the interface from where the switch derives the IP address that it uses to fill the source field in all outbound PIM SM register packets. The **ip pim register-source** command specifies this interface.

By default, the source field is filled with the IP address from the interface associated with the best route to the RP. The **show ip pim register-source** command does not return a value when the source field is filled with the default value.

Command Mode

EXEC

Command Syntax

```
show ip pim register-source
```

Example

- This command displays the register-source interface.

```
switch>show ip pim register-source
Ethernet22
switch>
```

show ip pim rp

The **show ip pim rp** command displays active rendezvous points (RPs) that are cached with associated multicast routing entries.

Command Mode

EXEC

Command Syntax

show ip pim rp

Example

- This command displays the active RPs.

```
switch>show ip pim rp
show ip pim rp
The PIM RP Set
Group: 224.0.0.0/4
  RP: 10.1.2.3
    Uptime: 00:05:12, Expires: never, Priority: 1 Override: 1
```

show ip pim rp-candidate

The **show ip pim rp-candidate** command displays the rendezvous point (RP) that is used for a specified multicast group.

Command Mode

EXEC

Command Syntax

```
show ip pim rp-candidate
```

Example

- This command configures the switch's candidate-RP information.

```
switch>show ip pim rp-candidate
Candidate RP information
  Candidate RP Address: 10.0.12.2
  CRP Holdtime: 150 seconds
    Group 224.2.0.0/16 Priority 2
```

show ip pim rp-hash

The **show ip pim rp-hash** displays the rendezvous point (RP) for a specified multicast group.

Command Mode

EXEC

Command Syntax

show ip pim rp-hash *ipv4_addr*

Parameters

- *ipv4_addr* multicast group IPv4 address.

Example

- This command displays the RP that the switch uses for multicast group 224.1.0.0.

```
switch>show ip pim rp-hash 224.1.0.0
RP 10.1.2.3
```

show ip pim upstream joins

The **show ip pim rp upstream joins** command displays the join messages that the switch is scheduled to send.

Command Mode

EXEC

Command Syntax

show ip pim upstream joins

Example

- This command displays the list of join messages the switch is scheduled to send. The example only displays the first two messages.

```
switch>show ip pim upstream joins

------------- show ip pim upstream joins -------------

Neighbor address: 10.1.1.1
Via interface: 10.1.1.2
Next message in 1 seconds
  Group: 239.10.10.3
    Joins:
      14.25.1.1/32 SPT
    Prunes:
      No prunes included
Neighbor address: 10.1.1.6
Via interface: 10.1.1.5
Next message in 1 seconds
  Group: 239.14.1.69
    Joins:
      17.105.14.3/32 SPT
    Prunes:
      No prunes included
```

Chapter 24

SNMP

This chapter describes the Arista switch SNMP agent and contains these sections:

- Section 24.1: SNMP Introduction
- Section 24.2: SNMP Conceptual Overview
- Section 24.3: Configuring SNMP
- Section 24.4: SNMP Commands

24.1 SNMP Introduction

Arista Networks switches support many standard SNMP MIBs, making it easier to integrate these platforms into existing network management infrastructures. With only a few configurations, many public domain and commercially available network management tools can quickly manage Arista switches out of the box. Support of SNMP V2 groups and views and V3 security allow network managers to tune switch monitoring to match the administration policy of the IT organization.

24.2 SNMP Conceptual Overview

Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a standardized framework and a common language to monitor and manage network devices.

24.2.1 SNMP Structure

The SNMP framework has three parts:

- **SNMP manager**: The SNMP manager controls and monitors network host activities and is typically part of a Network Management System (NMS).
- **SNMP agent**: The SNMP agent is the managed device component that manages and reports device information to the manager.
- **Management Information Base (MIB)**: The MIB stores network management information, which consists of collections of managed objects. Within the MIB are collections of related objects, defined in MIB modules.

  Table 24-1 lists the MIBs that the switch supports.

Feature	7100 Series	7500 Series	7048	7050 Series
All MIB support is read-only unless otherwise noted.				
SNMPv2, SNMPv3	YES	YES	YES	YES
RFC 3635 EtherLike-MIB (obsoletes RFCs 1650, 2358, 2665)	YES	YES	YES	YES
RFC 3418 SNMPv2-MIB (obsoletes RFCs 1450, 1907)	YES	YES	YES	YES
RFC 2863 IF-MIB (obsoletes RFCs 1229, 1573, 2233) (ifAdminStatus and ifAlias are writeable)	YES	YES	YES	YES
RFC 2864 IF-INVERTED-STACK-MIB	YES	YES	YES	YES
RFC 2096 IP-FORWARD-MIB (obsoletes RFC 1354)	YES	YES	YES	YES
ARISTA-SW-IP-FORWARD-MIB (IPv4 only)	YES	YES	YES	YES
RFC 4363 Q-BRIDGE-MIB (dot1qPvid and dot1qPortAcceptableFrameTypes are writeable for ports in switchport access or trunk mode)	YES	YES	YES	YES
RFC 4188 BRIDGE-MIB	YES	YES	YES	YES
ARISTA-BRIDGE-EXT-MIB	YES	YES	YES	YES
RFC 2013 UDP-MIB (obsoletes RFC 1213)	YES	YES	YES	YES
RFC 2012 TCP-MIB (obsoletes RFC 1213)	YES	YES	YES	YES
RFC 2011 IP-MIB (obsoletes RFC 1213) ipAddressTable and ipAddressPrefixTable from RFC 4293	YES	YES	YES	YES
HOST-RESOURCES-MIB	YES	YES	YES	YES
LLDP-MIB	YES	YES	YES	YES
LLDP-EXT-DOT1-MIB	YES	YES	YES	YES
LLDP-EXT-DOT3-MIB	YES	YES	YES	YES
ENTITY-MIB	YES	YES	YES	YES
ENTITY-SENSOR-MIB	YES	YES	YES	YES
ENTITY-STATE-MIB	YES	YES	YES	YES
RMON-MIB (rmonEtherStatsGroup)	YES	YES	YES	YES
RMON2-MIB (rmon1EthernetEnhancementGroup)	YES	YES	YES	YES
HC-RMON-MIB (etherStatsHighCapacityGroup)	YES	YES	YES	YES
RFC 3636 MAU-MIB (ifMauDefaultType and ifMauAutoNegStatus are writeable)	YES	YES	YES	YES
ARISTA-ACL-MIB	YES	YES	YES	YES

Table 24-1 SNMP Feature Support

The agent and MIB reside on the switch. Enabling the SNMP agent requires the definition of the manager-agent relationship. The agent contains MIB variables whose values the manager can request or change. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to manager requests for information.

A manager can send the agent requests to get and set MIB values. The agent can respond to these requests. Independent of this interaction, the agent can send unsolicited messages to the manager to notify the manager of network conditions.

This chapter discusses enabling the SNMP agent on an Arista switch and controlling notification transmissions from the agent. Information on using SNMP management systems is available in the appropriate documentation for the corresponding NMS application.

24.2.2 SNMP Notifications

SNMP notifications are messages, sent by the agent, to inform managers of an event or a network condition. A ***trap*** is an unsolicited notification. An ***inform*** (or inform request) is a trap that includes a request for a confirmation that the message is received. Events that a notification can indicate include improper user authentication, restart, and connection losses.

Traps are less reliable than informs because the receiver does not send any acknowledgment. However, traps are often preferred because informs consume more switch and network resources. A trap is sent only once and is discarded as soon as it is sent. An inform request remains in memory until a response is received or the request times out. An inform may be retried several times, increasing traffic and contributing to higher network overhead.

Table 24-2 lists the SNMP traps that the switch supports.

Feature	7100 Series	7500 Series	7048	7050 Series
RFC 2863 IF-MIB (linkUp, linkDown)	YES	YES	YES	YES
LLDP-MIB (lldpRemTablesChange)	YES	YES	YES	YES
RFC 3418 SNMPv2-MIB (coldStart)	YES	YES	YES	YES
NET-SNMP-AGENT-MIB (nsNotifyRestart)	YES	YES	YES	YES
ENTITY-MIB (entConfigChange)	YES	YES	YES	YES
ENTITY-STATE-MIB (entStateOperEnabled, entStateOperDisabled)	YES	YES	YES	YES

Table 24-2 Supported SNMP Traps

24.2.3 SNMP Versions

Arista switches support the following SNMP versions:

- **SNMPv1**: The Simple Network Management Protocol, defined in RFC 1157. Security is based on community strings.
- **SNMPv2c**: Community-string based Administrative Framework for SNMPv2, defined in RFC 1901, RFC 1905, and RFC 1906. SNMPv2c uses the community-based security model of SNMPv1.
- **SNMPv3**: Version 3 is an interoperable standards-based protocol defined in RFCs 2273 to 2275. SNMPv3 provides secure access to devices by authenticating and encrypting packets.

  The security features provided in SNMPv3 are as follows:

  — ***Message integrity***: Ensures packets are not tampered with in transit.
  — ***Authentication***: Determines the message is received from a valid source.
  — ***Encryption***: Scrambling packet contents to prevent an unauthorized source from learning it.

Both SNMPv1 and SNMPv2c use a community-based form of security. The community of managers able to access the agent MIB is controlled by a password.

SNMPv2c support includes a bulk retrieval mechanism and more detailed error message reporting. The bulk retrieval mechanism supports the retrieval of tables and large quantities of information, minimizing the number of round-trips required. SNMPv2c error handling includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1. SNMPv2c error return codes report error type.

SNMPv3 is a security model which defines an authentication strategy that is configured for a user and the group in which the user resides. A security level is the permitted level of security within the model. A combination of a security model and a security level determines the security mechanism employed to handle an SNMP packet.

24.3 Configuring SNMP

This section describes the steps that configure the switch SNMP agent to communicate with an SNMP manager, including the following:

- **Enabling and Disabling SNMP**
- **Configuring Community Access Control**
- **Configuring SNMP Parameters**
- **Configuring the Agent to Send Notifications**
- **Extending the SNMP Agent Through Run Time Scripts**

24.3.1 Enabling and Disabling SNMP

SNMP is enabled by issuing any **snmp-server community** or **snmp-server user** command.

The **no snmp-server** command disables SNMP agent operation by removing all non-default **snmp-server** commands from ***running-config***.

24.3.2 Specifying the SNMP VRF

By default, SNMP uses the main VRF for communication with SNMP servers. The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in the main VRF.

SNMP may only be enabled in one VRF at a time. Enabling SNMP in multiple VRFs disables SNMP on the switch. To enable SNMP in a user defined VRF, first disable it in main VRF by using the **no** form of the **snmp-server vrf** command, then enable SNMP in the user defined VRF.

24.3.3 Configuring Community Access Control

SNMP community strings authenticate access to MIB objects and function as embedded passwords. The community string serves as a password that permits an SNMP manager to access the agent on the switch. A Network Management System (NMS) can access the switch only if its community string matches at least one of the switch's community strings.

The **snmp-server community** command configures the community string.

Example

- This command adds the community string ***lab_1*** to provide read-only access to the switch agent.

```
switch(config)#snmp-server community lab_1 ro
```

Community statements can reference views to limit MIB objects that are available to a manager. A view is a community string object that specifies a subset of MIB objects. The **snmp-server view** command configures the community string.

Example

- These commands create a view that includes all objects in the ***system*** group except for those in ***system.2***.

```
switch(config)#snmp-server view sys-view system include
switch(config)#snmp-server view sys-view system.2 exclude
```

- This command adds the community string ***lab_1*** to provide read-only access to the switch agent for the previously defined view.

```
switch(config)#snmp-server community lab_1 sys-view
```

24.3.4 Configuring SNMP Parameters

This section describes these SNMP parameter configuration tasks:

- **Configuring the Engine ID**
- **Configuring the Group**
- **Configuring the User**
- **Configuring the Host**
- **Enabling Link Trap Generation**
- **Configuring the Chassis-id String**
- **Configuring the Contact String**
- **Configuring the Location String**

Configuring the Engine ID

The **snmp-server engineID remote** command configures the name for the local or remote Simple Network Management Protocol (SNMP) engine. An SNMP engine ID is a name for the local or remote SNMP engine.

A remote agent's engine ID must be configured before remote users for that agent are configured. User authentication and privacy digests are derived from the engine ID and user passwords. The configuration command fails if the remote engine ID is not configured first.

Important When the remote engine ID is changed, all user passwords associated with the engine must be reconfigured.

Example

- This command configures DC945798CAB4 as the name of the remote SNMP engine located at 12.23.104.25, port socket 162.

```
switch(config)#snmp-server engineID remote 10.23.104.25 udp-port DC945798CA
```

Configuring the Group

An SNMP group is a table that maps SNMP users to SNMP views. The **snmp-server group** command configures a new SNMP group.

Example

- This command configures ***normal_one*** as an SNMPv3 group (authentication and encryption) that provides access to the ***all-items*** read view.

```
switch(config)#snmp-server group normal_one v3 priv read all-items
```

Configuring the User

An SNMP user is a member of an SNMP group. The **snmp-server user** command adds a new user to an SNMP group and configures that user's parameters. To configure a remote user, specify the IP address or port number of the device where the user's remote SNMP agent resides.

Example

- This command configures the local SNMPv3 user ***tech-1*** as a member of the SNMP group ***tech-sup***.

```
switch(config)#snmp-server user tech-1 tech-sup v3
```

- This command configures the remote SNMPv3 user ***tech-2*** as a member of the SNMP group ***tech-sup***. The remote user is on the agent located at 13.1.1.4.

```
switch(config)#snmp-server user tech-2 tech-sup remote 13.1.1.4 v3
```

Configuring the Host

The **snmp-server host** command specifies the recipient of a SNMP notification. An SNMP host is the recipient of an SNMP trap operation. The **snmp-server host** command sets the community string if it was not previously configured.

Example

- This command adds a v2c inform notification recipient at 12.15.2.3 using the community string ***comm-1***.

```
switch(config)#snmp-server host 12.15.2.3 informs version 2c comm-1
switch(config)#
```

Enabling Link Trap Generation

The **snmp trap link-status** command enables SNMP link trap generation on the configuration mode interface. SNMP link trap generation is enabled by default. If SNMP link trap generation was previously disabled, this command removes the corresponding **no snmp link-status** statement from the configuration.

Example

- This command disables SNMP link trap generation on the Ethernet 5 interface.

```
switch(config-if-Et5)#no snmp trap link-status
switch(config-if-Et5)#
```

Configuring the Chassis-id String

The chassis ID string is typically set to the serial number of the switch. The SNMP manager uses this string to associate all data retrieved from the switch with a unique identifying label. Under normal operating conditions, editing the chassis ID string contents is unnecessary.

The **snmp-server chassis-id** command configures the chassis ID string. The default chassis ID string is the serial number of the switch. The **show snmp** command displays the chassis ID.

Example

- This command configures *xyz-1234* as the chassis-ID string, then displays the result.

```
switch(config)#snmp-server chassis-id xyz-1234
switch(config)#show snmp
Chassis: xyz-1234                                    <---chassis ID
8 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    8 Number of requested variables
    0 Number of altered variables
    4 Get-request PDUs
    4 Get-next PDUs
    0 Set-request PDUs
21 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    8 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to taccon.162
SNMP agent enabled
switch(config)#
```

Configuring the Contact String

The SNMP contact string is information text that typically displays the name of a person or organization associated with the SNMP agent.

The **snmp-server contact** command configures the system contact string. The contact string is displayed by the **show snmp** and **show snmp contact** commands.

Example

- These commands configure *Bonnie H at 3-1470* as the contact string, then displays the result.

```
switch(config)#snmp-server contact Bonnie H at 3-1470
switch(config)#show snmp
Chassis: xyz-1234
Contact: Bonnie H at
3-1470                                                    <---contact string
8 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    8 Number of requested variables
    0 Number of altered variables
    4 Get-request PDUs
    4 Get-next PDUs
    0 Set-request PDUs
24 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    8 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to taccon.162
SNMP agent enabled
switch(config)#
```

Configuring the Location String

The location string typically provides information about the physical location of the SNMP agent. The **snmp-server location** command configures the system location string. By default, the system location string is not set.

Example

- These commands configure *lab-25* as the location string, then displays the result.

```
switch(config)#snmp-server location lab_25
switch(config)#show snmp location
Location: lab_25
```

24.3.5 Configuring the Agent to Send Notifications

The following steps are required to set up the SNMP agent to send notifications:

1. Configure the remote engine ID.
2. Configure the group.
3. Configure the user.
4. Configure the host.
5. Enable link trap generation on the interfaces.

Section 24.3.4 describes each of these tasks.

24.3.6 Extending the SNMP Agent Through Run Time Scripts

The switch supports the execution of user supplied scripts to service portions of the OID space.

Scripts run under one of two operational modes:

- Normal: scripts run over an indefinite period to process subsequent objects after the initial request. Maintaining an executing script avoids startup and connection delay each time an object requires processing. is processed; the can also cache its results)
- One-shot mode: scripts process a single object, then terminates execution.

Normal extension scripts are conceptually multithreaded: one thread collects data and the other thread is ready to communicate with snmpd. One-shot scripts process a single object, running once and exiting. Startup and data collection overhead is required for each request.

In both modes, the SNMP server is blocked from serving other requests when waiting for script responses.

The **snmp-server extension** command configures the execution of user supplied scripts to service portions of the OID space.

Example

- This command specifies the file ***example.sh***, located in flash as the script file that services the specified OID space in normal mode.

```
switch(config)#snmp-server extension .1.3.6.1.4.1.8072.2 flash:example.sh
```

24.3.6.1 Normal Script Behavior

The first time the SNMP server requires a script result, it launches it with no arguments. The server communicates with the script through stdin/stdout. Before each request, the script is the string ***PING\n*** on **stdin**. The expected response is printing ***PONG\n*** to **stdout**.

GET and GETNEXT Requests

For GET and GETNEXT requests, the script is passed two lines on stdin, the command (get or getnext) and the requested OID. The expected response from the script is the printing of three lines to stdout: the OID for the result varbind, the TYPE and the VALUE itself.

Table 24-3 lists legal TYPE values and resulting VALUE encodings. If the command cannot return an appropriate varbind, it should print print "NONE\n" to stdout and continue running; this results in an SNMP ***noSuchName*** error or a ***noSuchInstance*** exception.

Example

These commands are example GET and GETNEXT transactions:

```
--> get
--> .1.3.6.1.4.1.8072.2.255.1.0

<-- .1.3.6.1.4.1.8072.2.255.1.0
<-- string
<-- Sales Objectives

--> getnext
--> .1.3.6.1.4.1.8072.2.255.1.0

<-- .1.3.6.1.4.1.8072.2.255.2.1.2.1
<-- integer
<-- 17

--> getnext
--> .1.3.6.1.4.1.8072.2.255.2.1.2.1
```

```
<-- .1.3.6.1.4.1.8072.2.255.2.1.3.1
<-- objectid
<-- .1.3.6.1.4.1.8072.2.255.99

--> getnext
--> .1.3.6.1.4.1.8072.2.256

<-- NONE

--> get
--> .1.3.6.1.4.1.8072.2.255.2.1.2.2

<-- NONE
```

Type string	SNMP type	Encoding for script
integer	Integer32	integer
unsigned	Unsigned32	integer
gauge	Gauge32	integer
counter	Counter32	integer
counter64	Counter64	integer
timetick	TimeTicks	integer
ipaddress	IpAddress	a.b.c.d
objectid	ObjectID	1.3.6.1.42.99.2468
octet	OctetString	hexadecimal string
opaque	Opaque	hexadecimal string
string	OctetString	ascii string

Table 24-3 Extension Script Type and Encoding

SET

For SET requests, script is passed three lines on stdin: the command (set), and the requested OID, and the type and value, both on the same line. If the assignment is successful, the expected script response is to print **DONE\n** to stdout. Errors should be indicated by writing one of the error strings described in Table 24-4 to stdout; the agent the generates the appropriate error response. In each case, the command should continue running.

authorization-error	no-access	too-big
bad-value	no-creation	undo-failed
commit-failed	no-such-name	wrong-type
gen-error	not-writable	wrong-length
inconsistent-name	read-only	wrong-encoding
inconsistent-value	resource-unavailable	wrong-value

Table 24-4 Set Request Error Strings

24.3.6.2 One Shot Script Behavior

The command should exit after it finishes processing a single object. Each request, and each varbind a single request, triggers a separate command invocation.

GET and GETNEXT

For each GET or GETNEXT request, the script is invoked once for each OID in the space that it serves. It receives two arguments: -g for GET or -n for GETNEXT, and the requested OID.

The expected script response is the response varbind as three separate lines printed to stdout: the result OID, the type, and the value.

If the command cannot return an appropriate varbind, then the script should exit without producing any output. This results in an SNMP ***noSuchName*** error, or a ***noSuchInstance exception***.

Possible reasons that a command would not return an appropriate varbind includes:

- the specified OID did not correspond to a valid instance for a GET request
- there were no following instances for a GETNEXT -

SET

A SET request results in the command being called with the arguments: -s, OID, TYPE and VALUE, where TYPE is one of the tokens listed Table 24-3, indicating the type of the value passed as the third parameter.

If the assignment is successful, the script is expected to exit without producing any output. Errors should be indicated by writing just the error name (Table 24-4); the agent generates the appropriate error response.

24.4 SNMP Commands

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Commands

- no snmp-server . . . . . Page 1117
- snmp-server chassis-id . . . . . Page 1129
- snmp-server community . . . . . Page 1130
- snmp-server contact . . . . . Page 1131
- snmp-server enable traps . . . . . Page 1132
- snmp-server engineID local . . . . . Page 1133
- snmp-server engineID remote . . . . . Page 1134
- snmp-server extension . . . . . Page 1135
- snmp-server group . . . . . Page 1136
- snmp-server host . . . . . Page 1137
- snmp-server location . . . . . Page 1139
- snmp-server source-interface . . . . . Page 1140
- snmp-server user . . . . . Page 1141
- snmp-server view . . . . . Page 1142
- snmp-server vrf . . . . . Page 1143

Interface Configuration Commands

- snmp trap link-status . . . . . Page 1144

Display Commands

- show snmp . . . . . Page 1118
- show snmp chassis . . . . . Page 1119
- show snmp community . . . . . Page 1120
- show snmp contact . . . . . Page 1121
- show snmp engineID . . . . . Page 1122
- show snmp group . . . . . Page 1123
- show snmp host . . . . . Page 1124
- show snmp location . . . . . Page 1125
- show snmp mib . . . . . Page 1126
- show snmp user . . . . . Page 1127
- show snmp view . . . . . Page 1128

no snmp-server

The **no snmp-server** and **default snmp-server** commands disable Simple Network Management Protocol (SNMP) agent operation by removing all **snmp-server** commands from ***running-config***.

SNMP is enabled with any **snmp-server community** or **snmp-server user** command.

Command Mode

Global Configuration

Command Syntax

```
no snmp-server
default snmp-server
```

Example

This command disables SNMP agent operation on the switch

```
switch(config)#no snmp-server
switch(config)#
```

show snmp

The **show snmp** command displays SNMP information including the SNMP counter status and the chassis ID string.

Command Mode

EXEC

Command Syntax

show snmp

Example

This command displays SNMP counter status, the chassis ID, the previously configured location string, logging status and destination, and the VRF in which the SNMP agent is operating.

```
switch>show snmp
Chassis: JFL08320162
Location: 5470ga.dc
2329135 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    38132599 Number of requested variables
    0 Number of altered variables
    563934 Get-request PDUs
    148236 Get-next PDUs
    0 Set-request PDUs
2329437 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    2329135 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to 172.22.22.20.162
SNMP agent configured in VRFs: main
SNMP agent enabled in main VRF
switch>
```

show snmp chassis

The **show snmp chassis** command displays the Simple Network Management Protocol (SNMP) server serial number or the chassis ID string configured by the **snmp-server chassis-id** command.

Command Mode

EXEC

Command Syntax

```
show snmp chassis
```

Example

This command displays the chassis ID string.

```
switch>show snmp chassis
Chassis: JFL08320162
switch>
```

show snmp community

The **show snmp community** command displays the Simple Network Management Protocol (SNMP) community access strings configured by the **snmp-server community** command.

Command Mode

EXEC

Command Syntax

```
show snmp community
```

Example

This command displays the list of community access strings configured on the switch.

```
switch>show snmp community

Community name: public
switch>
```

show snmp contact

The **show snmp contact** command displays the Simple Network Management Protocol (SNMP) system contact string configured by the **snmp-server contact** command. The command has no effect if a contact string was not previously configured.

Command Mode

EXEC

Command Syntax

```
show snmp contact
```

Example

This command displays the contact string contents.

```
switch>show snmp contact
Contact: John Smith
switch>
```

show snmp engineID

The **show snmp engineID** command displays the identification of the local Simple Network Management Protocol (SNMP) engine and of all remote engines that are configured on the switch.

Command Mode

EXEC

Command Syntax

```
show snmp engineID
```

Example

This command displays the ID of the local SNMP engine.

```
switch>show snmp engineid
Local SNMP EngineID: f5717f001c730436d700
switch>
```

show snmp group

The **show snmp group** command displays the names of configured SNMP groups along with the security model, and view status of each group.

Command Mode

EXEC

Command Syntax

```
show snmp group
```

Field Descriptions

- **groupname** name of the SNMP group.
- **security model** security model used by the group: *v1*, *v2c*, or *v3*.
- **readview** string identifying the group's read view. Refer to **show snmp view**.
- **writeview** string identifying the group's write view.
- **notifyview** string identifying the group's notify view.

  The notify view indicates the group for SNMP notifications, and corresponds to the ***notify-view*** specified in the **snmp-server group** command.

Example

This command displays the groups configured on the switch.

```
switch>show snmp group
groupname : normal                          security model:v3 priv
readview  : all                             writeview: <no writeview specified>
notifyview: <no notifyview specified>

switch>
```

show snmp host

The **show snmp host** command displays the recipient details for Simple Network Management Protocol (SNMP) notification operations. Details that the command displays include IP address and port number of the Network Management System (NMS), notification type, and SNMP version.

Command Mode

EXEC

Command Syntax

```
show snmp host
```

Field Descriptions

- **Notification host** IP address of the host for which the notification is generated.
- **udp-port** port number.
- **type** notification type.
- **user** access type of the user for which the notification is generated.
- **security model** SNMP version used to send notifications.
- **traps** details of the notification generated.

Example

This command displays the hosts configured on the switch.

```
switch>show snmp host
Notification host: 172.22.22.20    udp-port: 162   type: trap
user: public                       security model: v2c

switch>
```

show snmp location

The **show snmp location** command displays the Simple Network Management Protocol (SNMP) system location string. The **snmp-server location** command configures system location details. The command has no effect if a location string was not previously configured.

Command Mode

EXEC

Command Syntax

```
show snmp location
```

Example

This command displays the location string contents.

```
switch>show snmp location
Location: santa clara
switch>
```

show snmp mib

The **show snmp mib** command displays values associated with specified MIB object identifiers (OIDs) that are registered on the switch.

Command Mode

EXEC

Command Syntax

show snmp mib ***OBJECTS***

Parameters

- ***OBJECTS*** object identifiers for which the command returns data. Options include:
  - **get** *oid_1* [*oid_2 ... oid_x*] values associated with each listed OID.
  - **get-next** *oid_1* [*oid_2 ... oid_x*] values associated with subsequent OIDs relative to listed OIDs.
  - **table** *oid* table associated with specified OID.
  - **translate** *oid* object name associated with specified OID.
  - **walk** *oid* objects below the specified subtree.

Example

This command uses the get option to retrieve information about the sysORID.1 OID.

```
switch#show snmp mib get sysORID.1
SNMPv2-MIB::sysORID[1] = OID: TCP-MIB::tcpMIB
```

This commnd uses the get-next option to retrieve information about the OID that is after sysORID.8.

```
switch#show snmp mib get-next sysORID.8
SNMPv2-MIB::sysORDescr[1] = STRING: The MIB module for managing TCP
implementations
switch>show snmp location
Location: santa clara
switch>
```

show snmp user

The **show snmp user** command displays information about Simple Network Management Protocol (SNMP) users. Information that the command displays about each user includes their SNMP version, the engine ID of the host where they reside, and security information.

Command Mode

EXEC

Command Syntax

show snmp user

Example

This command displays information about the users configured on the switch.

```
switch>show snmp user

User name: test

Security model: v3
Engine ID: f5717f001c73010e0900
Authentication protocol: SHA
Privacy protocol: AES-128
Group name: normal
switch>
```

show snmp view

The **show snmp view** command displays the family name, storage type, and status of a Simple Network Management Protocol (SNMP) configuration and the associated MIB. SNMP views are configured with the **snmp-server view** command.

Command Mode

EXEC

Command Syntax

```
show snmp view
```

Field Descriptions

- ***First column*** view name.
- ***Second column*** name of the MIB object or family.
- ***Third column*** inclusion level of the specified family within the view.

Example

These commands configure an SNMP view, then displays that view.

```
switch(config)#snmp-server view sys-view system include
switch(config)#snmp-server view sys-view system.2 exclude
switch(config)#show snmp view
sys-view system - included
sys-view system.2 - excluded
```

snmp-server chassis-id

The **snmp-server chassis-id** command configures the chassis ID string. The default chassis ID string is the serial number of the switch. The the **show snmp** command displays the chassis ID.

The **no snmp-server chassis-id** and **default snmp-server chassis-id** commands restore the default chassis ID string by removing the **snmp-server chassis-id** command from the configuration.

Command Mode

Global Configuration

Command Syntax

```
snmp-server chassis-id id_text
no snmp-server chassis-id
default snmp-server chassis-id
```

Parameters

- *id_ext* chassis ID string

Example

These commands configure *xyz-1234* as the chassis-id string, then display the result.

```
switch(config)#snmp-server chassis-id xyz-1234
switch(config)#show snmp
Chassis: xyz-1234                                    <---chassis ID
8 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    8 Number of requested variables
    0 Number of altered variables
    4 Get-request PDUs
    4 Get-next PDUs
    0 Set-request PDUs
21 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    8 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to taccon.162
SNMP agent enabled
switch(config)#
```

snmp-server community

The **snmp-server community** command configures the community string. SNMP community strings authenticate access to MIB objects and function as embedded passwords. The Network Management System (NMS) must define a community string that matches at least one of the switch community strings to access the switch.

The **no snmp-server community** and **default snmp-server community** commands remove the community access string from the configuration.

Command Mode

Global Configuration

Command Syntax

```
snmp-server community string_text [MIB_VIEW] [ACCESS]
no snmp-server community string_text
default snmp-server community string_text
```

Parameters

- *string_text* community access string.
- ***MIB_VIEW*** community access availability. Options include
  - <no parameter> community string allows access to all objects.
  - **view** *view_name* community string allows access only to objects in the *view_name* view.
- ***ACCESS*** community access availability. Options include
  - <no parameter> read-only access (default setting)
  - **ro** read-only access
  - **rw** read-write access

Example

This command adds the community string lab_1 to provide read-only access to the switch agent.

```
switch(config)#snmp-server community lab_1 ro
switch(config)#
```

snmp-server contact

The **snmp-server contact** command configures the system contact string. The contact is displayed by the **show snmp** and **show snmp contact** commands.

The **no snmp-server contact** and **default snmp-server contact** commands remove the **snmp-server contact** command from the configuration.

Command Mode

Global Configuration

Command Syntax

```
snmp-server contact contact_string
no snmp-server contact
default snmp-server contact
```

Parameters

- *contact_string* system contact string.

Example

These commands configure *Bonnie H* as the contact string, then display the result.

```
switch(config)#snmp-server contact Bonnie H
switch(config)#show snmp
Chassis: xyz-1234
Contact: Bonnie H.
8 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    8 Number of requested variables
    0 Number of altered variables
    4 Get-request PDUs
    4 Get-next PDUs
    0 Set-request PDUs
24 SNMP packets output
    0 Too big errors
    0 No such name errors
    0 Bad value errors
    0 General errors
    8 Response PDUs
    0 Trap PDUs
SNMP logging: enabled
    Logging to taccon.162
SNMP agent enabled
switch(config)#
```

snmp-server enable traps

The **snmp-server enable traps** command enables the transmission of Simple Network Management Protocol (SNMP) notifications as traps or inform requests. This command enables both traps and inform requests for the specified notification types. The **snmp-server host** command specifies the notification type (traps or informs). Sending notifications requires at least one **snmp-server host** command.

The **snmp-server enable traps** and **no snmp-server enable traps** commands, without an MIB parameter, specifies the default notification trap generation setting for all MIBs. These commands, when specifying an MIB, controls notification generation for the specified MIB. The default snmp-server enable command resets notification generation to the default setting for the specified MIB.

Command Mode

Global Configuration

Command Syntax

```
snmp-server enable [trap_type]
no snmp-server enable [trap_type]
default snmp-server enable trap_type
```

Parameters

- *trap_type* controls the generation of informs or traps for the specified MIB:
  - <no parameter> controls notifications for MIBs not covered by specific commands.
  - **entity** entity-MIB modification notifications.
  - **lldp** LLDP-MIB.
  - **snmp** SNMP-v2-MIB.
  - **test** TEST-MIB.

Examples

These commands enables notification generation for all MIBs except spanning tree.

```
switch(config)#snmp-server enable traps
switch(config)#no snmp-server enable traps spanning-tree
switch(config)#
```

This command enables spanning-tree MIB notification generation, regardless of the default setting.

```
switch(config)#snmp-server enable traps spanning-tree
switch(config)#
```

This command resest the spanning-tree MIB notification generation to follow the default setting.

```
switch(config)#default snmp-server enable traps spanning-tree
switch(config)#
```

snmp-server engineID local

The **snmp-server engineID local** command configures the name for the local Simple Network Management Protocol (SNMP) engine. The default SNMP engineID is generated by the switch and is used when an engineID is not configured with this command. The **show snmp engineID** command displays the default or configured engine ID.

SNMPv3 authenticates users through security digests (MD5 or SHA) that are based on user passwords and the local engine ID. Passwords entered on the CLI are similarly converted, then compared to the user's security digest to authenticate the user.

Important Changing the local engineID value invalidates SNMPv3 security digests, requiring the reconfiguration of all user passwords.

The **no snmp-server engineID** and **default snmp-server engineID** commands restore the default engineID by removing the **snmp-server engineID** command from the configuration**.**

Command Mode

Global Configuration

Command Syntax

```
snmp-server engineID local engine_hex
no snmp-server engineID local
default snmp-server engineID
```

Parameters

- *engine_hex* the switch's name for the local SNMP engine (hex string).

  The string must consist of at least ten characters with a maximum of 64 characters.

Example

This command configures DC945798CAB4 as the name of the local SNMP engine.

```
switch(config)#snmp-server engineID local DC945798CAB4
switch(config)#
```

snmp-server engineID remote

The **snmp-server engineID remote** command configures the name of a Simple Network Management Protocol (SNMP) engine located on a remote device. The switch generates a default engineID; use the **show snmp engineID** command to view the configured or default engineID.

A remote engine ID is required when configuring an SNMPv3 inform to compute the security digest for authenticating and encrypting packets sent to users on the remote host. SNMPv3 authenticates users through security digests (MD5 or SHA) that are based on user passwords and the engine ID. Passwords entered on the CLI are similarly converted, then compared to the user's security digest to authenticate the user.

Important Changing the engineID value invalidates SNMPv3 security digests, requiring the reconfiguration of all user passwords.

The **no snmp-server engineID remote** and **default snmp-server engineID remote** commands remove the **snmp-server engineID remote** command from the configuration**.**

Command Mode

Global Configuration

Command Syntax

```
snmp-server engineID remote engine_addr [PORT] engine_hex
no snmp-server engineID remote engine_addr [PORT]
default snmp-server engineID remote engine_addr [PORT]
```

Parameters

- *engine_addr* location of remote engine (IP address or host name).
- ***PORT*** udp port location of the remote engine. Options include:
  - — <No parameter> port number 161 (default).
  - — **udp-port** *port_num* port number. Ranges from 0 to 65536.
- *engine_hex* the switch's name for the remote SNMP engine (hex string).

  The string must have at least ten characters and can contain a maximum of 64 characters.

Example

This command configures DC945798CA as the engineID of the remote SNMP engine located at 12.23.10.25, port socket 162.

```
switch(config)#snmp-server engineID remote 10.23.10.25 udp-port 162 DC945798CA
switch(config)#
```

snmp-server extension

The **snmp-server extension** command configures the execution of user supplied scripts to service portions of the OID space.

The **no snmp-server extension** and **default snmp-server extension** commands deletes the **snmp-server extension** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
snmp-server extension OID_space FILE_PATH [DURATION]
```

Parameters

- *OID_space* OID branch serviced by the script. Command format is numerical.
- ***FILE_PATH*** path and name of the script file. Options include:
  - **file:** file is located in the switch file directory.
  - **flash:** file is located in flash memory.
- ***DURATION*** the execution scope of the script.
  - <no parameter> script runs after initial request to process subsequent requests.
  - **one-shot** script processes a single object (runs once), then terminates.

Examples

- This command specifies the file ***example.sh***, located in flash, as the script file that services the listed OID space.

```
switch(config)#snmp-server extension .1.3.6.1.4.1.8072.2 flash:example.sh
```

snmp-server group

The **snmp-server group** command configures a new Simple Network Management Protocol (SNMP) group or modifies an existing group. An SNMP group is a data structure that user statements reference to map SNMP users to SNMP contexts and views, providing a common access policy to the specified users.

An SNMP context is a collection of management information items accessible by an SNMP entity. Each item of may exist in multiple contexts. Each SNMP entity can access multiple contexts. A context is identified by the EngineID of the hosting device and a context name.

The **no snmp-server group** and **default snmp-server group** commands delete the specified group by removing the corresponding **snmp-server group** command from the configuration.

Command Mode

Global Configuration

Command Syntax

```
snmp-server group group_name VERSION [CNTX] [READ] [WRITE] [NOTIFY]
no snmp-server group group_name VERSION
default snmp-server group group_name VERSION
```

Parameters

- *group_name* the name of the group.
- ***VERSION*** the security model used by the group.
  - **v1** SNMPv1. Uses a community string match for authentication.
  - **v2c** SNMPv2c. Uses a community string match for authentication.
  - **v3 no auth** SNMPv3. Uses a username match for authentication.
  - **v3 auth** SNMPv3. HMAC-MD5 or HMAC-SHA authentication.
  - **v3 priv** SNMPv3. HMAC-MD5 or HMAC-SHA authentication. AES or DES encryption.
- ***CNTX*** associates the SNMP group to an SNMP context.
  - <no parameter> command does not associate group with an SNMP context.
  - **context** *context_name* associates group with context specified by *context_name*.
- ***READ*** specifies read view for SNMP group.
  - <no parameter> command does not specify read view.
  - **read** *read_name* read view specified by *read_name (*string – maximum 64 characters).
- ***WRITE*** specifies write view for SNMP group.
  - <no parameter> command does not specify write view.
  - **write** *write_name* write view specified by *write_name (*string – maximum 64 characters).
- ***NOTIFY*** specifies notify view for SNMP group.
  - <no parameter> command does not specify notify view.
  - **notify** *notify_name* notify view specified by *notify_name (*string – maximum 64 characters).

Example

This command configures normal_one as SNMP version 3 group (authentication and encryption) that provides access to the ***all-items*** read view.

```
switch(config)#snmp-server group normal_one v3 priv read all-items
switch(config)#
```

snmp-server host

The **snmp-server host** command specifies the recipient of Simple Network Management Protocol (SNMP) notifications. Recipients are denoted by host location and community string. The command also specifies the type of SNMP notifications that are sent: a ***trap*** is an unsolicited notification; an ***inform*** is a trap that includes a request for a confirmation that the message is received.

The configuration can contain multiple statements to the same host location with different community strings. For instance, a configuration can simultaneously contain all of the following:

- snmp-server host host-1 version 2c comm-1
- snmp-server host host-1 informs version 2c comm-2
- snmp-server host host-1 version 2c comm-3 udp-port 666
- snmp-server host host-1 version 3 auth comm-3

The **no snmp-server host** and **default snmp-server host** commands remove the specified host by deleting the corresponding **snmp-server host** statement from the configuration. When removing a statement, the host (address and port) and community string must be specified.

Command Mode

Global Configuration

Command Syntax

```
snmp-server host host_id [VRF_INST] [MESSAGE] [VERSION] comm_str [PORT]
no snmp-server host host_id [VRF_INST] [MESSAGE] [VERSION] comm_str [PORT]
default snmp-server host host_id [VRF_INST] [MESSAGE] [VERSION] comm_str [PORT]
```

Parameters

- *host_id* hostname or IP address of the targeted recipient.
- ***VRF_INST*** specifies the VRF instance being modified.
  - <no parameter> changes are made to the main VRF.
  - **vrf** *vrf_name* changes are made to the specified user defined VRF.
- ***MESSAGE*** message type that is sent to the host.
  - <no parameter> sends SNMP traps to host (default).
  - **informs** sends SNMP informs to host.
  - **traps** sends SNMP traps to host.
- ***VERSION*** SNMP version. Options include:
  - <no parameter> SNMPv2c (default).
  - **version 1** SNMPv1; option not available with informs.
  - **version 2c** SNMPv2c.
  - **version 3 noauth** SNMPv3; enables user-name match authentication.
  - **version 3 auth** SNMPv3; enables MD5 and SHA packet authentication.
  - **version 3 priv** SNMPv3. HMAC-MD5 or HMAC-SHA authentication. AES or DES encryption.
- *comm_str* community string (used as password) sent with the notification operation.

  Although this string can be set with the **snmp-server host** command, the preferred method is defining it with the **snmp-server community** command prior to using this command.
- ***PORT*** port number of the host.
  - <no parameter> socket number set to 162 (default)
  - **udp-port** *p-name* socket number specified by *p-name*

Guidelines

For traps and informs to be sent, the host location must be accessible through an interface in the main VRF.

Example

This command adds a version 2c inform notification recipient.

```
switch(config)#snmp-server host 12.15.2.3 informs version 2c comm-1
switch(config)#
```

snmp-server location

The **snmp-server location** command configures the system location string. By default, no system location string is set.

The **no snmp-server location** and **default snmp-server location** commands delete the location string by removing the **snmp-server location** command from the configuration.

Command Mode

Global Configuration

Command Syntax

```
snmp-server location node_locate
no snmp-server location
default snmp-server location
```

Parameters

- *node_locate* system location information (string).

Example

These commands configure *lab-east* as the location string, then displays the result.

```
switch(config)#snmp-server location lab_east
switch(config)#show snmp location
Location: lab_east
```

snmp-server source-interface

The **snmp-server source-interface** command specifies the interface from which a Simple Network Management Protocol (SNMP) trap originates the informs or traps.

The **no snmp-server source-interface** and **default snmp-server source-interface** commands remove the inform or trap source assignment by removing the **snmp-server source-interface** command from **running-config**.

Command Mode

Global Configuration

Command Syntax

```
snmp-server source-interface INTERFACE
no snmp-server source-interface
default snmp-server source-interface
```

Parameters

- *INTERFACE* Interface type and number. Values include
  - **ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **loopback** *l_num* Loopback interface specified by *l_num*.
  - **management** *m_num* Management interface specified by *m_num*.
  - **port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **vlan** *v_num* VLAN interface specified by *v_num*.

Example

This command configures the Ethernet 1 interface as the source of SNMP traps and informs.

```
switch(config)#snmp-server source-interface ethernet 1
```

snmp-server user

The **snmp-server user** command adds a user to a Simple Network Management Protocol (SNMP) group or modifies an existing user's parameters.

To configure a remote user, specify the IP address or port number of the device where the user's remote SNMP agent resides. A remote agent's engine ID must be configured before remote users for that agent are configured. A user's authentication and privacy digests are derived from the engine ID and the user's password. The configuration command fails if the remote engine ID is not configured first.

The **no snmp-server user** and **default snmp-server user** commands remove the user from an SNMP group by deleting the user command from the configuration.

Command Mode

Global Configuration

Command Syntax

```
snmp-server user user_name group_name [AGENT] VERSION [ENGINE][SECURITY]
no snmp-server user user_name group_name [AGENT] VERSION
default snmp-server user user_name group_name [AGENT] VERSION
```

Parameters

- *user_name* name of the user on the host that connects to the agent.
- *group_name* name of the group to which the user is associated.
- ***AGENT*** location of the host connecting to the SNMP agent. Configuration options include:
  - — <no parameter> local SNMP agent.
  - — **remote** *addr* [**udp-port** *p_num*] remote SNMP agent location (IP address, udp port).

    addr denotes the IP address*; p_num* denotes the udp port socket. (default port is 162).
- ***VERSION*** SNMP version; options include:
  - — **v1** SNMPv1.
  - — **v2c** SNMPv2c.
  - — **v3** SNMPv3; enables user-name match authentication.
- ***ENGINE*** engine ID used to localize passwords. Available only if ***VERSION*** is **v3**.
  - — <no parameter> Passwords localized by SNMP copy specified by *agent*.
  - — **localized** *engineID* octet string of engineID.
- ***SECURITY*** Specifies authentication and encryption levels. Available only if ***VERSION*** is **v3**. Encryption is available only when authentication is configured.
  - — <no parameter> no authentication or encryption.
  - — **auth** ***a_meth*** *a_pass* [**priv** ***e_meth*** *e_pass*] authentication and encryption parameters.

    a-meth authentication method: options are **md5** (HMAC-MD5-96) and **sha** (HMAC-SHA-96).
    a-pass authentication string for users receiving packets.
    e-meth encryption method: tions are **aes** (AES-128) and **des** (CBC-DES).
    e-pass encryption string for the users sending packets.

Example

This command configures the remote SNMP user ***tech-1*** to the ***tech-sup*** SNMP group.

```
switch(config)#snmp-server user tech-1 tech-sup remote 10.1.1.2 v3
```

snmp-server view

The **snmp-server view** command creates or updates a view entry.

An SNMP view defines a subset of objects from an MIB. Every SNMP access group specifies views, each associated with read or write access rights, to allow or limit the group's access to MIB objects.

The **no snmp-server view** command deletes a view entry by removing the corresponding **snmp-server view** command from the ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
snmp-server view view_name family_name INCLUSION
```

Parameters

- *view_name* Label for the view record that the command updates or creates. Other commands reference the view with this label.
- *family_name* name of the MIB object or family.

  MIB objects and MIB subtrees can be identified by name or by the numbers representing the position of the object or subtree in the MIB hierarchy.
- ***INCLUSION*** inclusion level of the specified family within the view. Options include:
  - **include** view includes the specified subtree.
  - **exclude** view excludes the specified subtree.

Example

These commands create a view named ***sys-view*** that includes all objects in the ***system*** subtree except for those in ***system.2***.

```
switch(config)#snmp-server view sys-view system include
switch(config)#snmp-server view sys-view system.2 exclude
```

snmp-server vrf

The **snmp-server vrf** command enables SNMP in the specified VRF. By default, SNMP is enabled in **main** VRF.

- User defined VRFs: The **no snmp-server vrf** command disables SNMP in the specified VRF by removing the corresponding **snmp-server vrf** command from the ***running-config***.
- Main VRF: The **no snmp-server vrf** command disables SNMP in the VRF by adding **no snmp-server vrf main** statement to ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
snmp-server vrf vrf_name
no snmp-server vrf vrf_name
default snmp-server vrf vrf_name
```

Parameters

- *vrf_name* The VRF in which SNMP is enabled. The keyword **main** specifies the default VRF.

Guidelines

SNMP may only be enabled in one VRF at a time. Enabling SNMP in multiple VRFs disables SNMP on the switch. To enable SNMP in a user defined VRF, first disable it in VRF **main** with the **no** form of the command.

The switch can only send SNMP traps and informs if the host that has been configured to receive them is accessible through an interface in **main** VRF.

Example

These commands disable SNMP in main VRF, then enable it in the user defined VRF named **magenta**.

```
switch(config)#no snmp-server vrf main
switch(config)#snmp-server vrf magenta
switch(config)#
```

snmp trap link-status

The **snmp trap link-status** command enables Simple Network Management Protocol (SNMP) link-status trap generation on the configuration mode interface. The generation of link-status traps is enabled by default. If SNMP link-trap generation was previously disabled, this command removes the corresponding **no snmp link-status** statement from the configuration to re-enable link-trap generation.

The **no snmp trap link-status** command disables SNMP link trap generation on the configuration mode interface.

The **snmp trap link-status** and **default snmp trap link-status** commands restore the default behavior by removing the **no snmp trap link-status** command from ***running-config***. Only the **no** form of this command is visible in ***running-config***.

Command Mode

Interface Configuration (Ethernet, Loopback, Management, Port-Channel, VLAN)

Command Syntax

```
snmp trap link-status
no snmp trap link-status
default snmp trap link-status
```

Guidelines

The switch can only generate SNMP traps and informs if SNMP is enabled in main VRF. Enable or disable SNMP in a VRF with the **snmp-server vrf** command.

SNMP may only be enabled in one VRF at a time. Enabling SNMP in multiple VRFs disables SNMP on the switch. To enable SNMP in a user defined VRF, first disable it in main VRF with the **no** form of the **snmp-server vrf** command.

Example

This command disables SNMP link trap generation on the Ethernet 5 interface.

```
switch(config-if-Et5)#no snmp trap link-status
switch(config-if-Et5)#
```

Chapter 25

Latency Analyzer (LANZ)

Arista Networks' Latency Analyzer (LANZ) is a family of EOS features that provide enhanced visibility into network dynamics, particularly in areas related to the delay packets experience through the network. The LANZ feature is available on the FM4000 and Petra switch platforms.

This chapter describes the purpose, behavior, and configuration of LANZ features. Topics covered by this chapter include:

- Section 25.1: Introduction to LANZ
- Section 25.2: LANZ Overview
- Section 25.3: Configuring LANZ
- Section 25.4: LANZ Commands

25.1 Introduction to LANZ

LANZ tracks interface congestion and queuing latency with real-time reporting. With LANZ application layer event export, external applications can predict impending congestion and latency. This enables the application layer to make traffic routing decisions with visibility into the network layer.

With LANZ, network operations teams and administrators have near real-time visibility into the network, enabling early detection of microbursts. LANZ continually monitors congestion, allowing for rapid detection of congestion and sending of application-layer messages.

25.2 LANZ Overview

LANZ monitors output queue lengths to provide congestion information for individual interfaces. This allows for more detailed analysis of congestion events, and allows identification of potential latency problems before they arise.

Output queues for each port are monitored, and information about queue congestion events can be accessed in the form of syslog messages, reports, or streaming.

25.2.1 LANZ Monitoring Mechanism

LANZ provides congestion data by continuously monitoring each port's output queue lengths. When the length of an output queue exceeds the upper threshold for that port, LANZ generates an over-threshold event. LANZ continues to report an over-threshold state every 800 microseconds until all queue lengths for that port pass below the lower threshold.

25.2.2 LANZ Logging

Over-threshold events generated by LANZ can be logged as syslog messages. Log messages are generated for events on all ports, at a maximum rate of one message per second per interface. The interval between messages can be configured globally.

Log messages indicate the time of the event, the interface affected, the threshold set for that interface, and the actual number of entries in the port's queue.

25.2.3 LANZ Reporting

Detailed LANZ data can be viewed through the CLI or exported as a CSV-formatted report.

A circular FIFO event buffer is dynamically shared by all interfaces. When an interface begins generating LANZ over-threshold events it can fill all available buffer space. However, each interface is guaranteed sufficient resources for a miminum of 500 entries.

25.2.4 LANZ Streaming

External client applications can also receive congestion event information as a data stream. The switch can stream LANZ data to up to 100 clients via TCP through port 50001. Streamed data is in Google protocol buffer format, and includes both over-threshold events and LANZ configuration information.

25.2.5 Platforms

The LANZ feature is available on the FM4000 and Petra switch platforms. To determine the switch platform, enter **show platform ?** at the prompt.

Settings and capabilities differ slightly between the platforms:

- The Petra chip measures threshold values in bytes; the FM4000 chip measures threshold values in segments.
- Only the FM4000 chip allows configuration of both upper and lower threshold values.
- Only the FM4000 chip support LANZ data streaming.
- While the FM4000 chip monitors congestions events for all queues, the Petra chip only monitors the most congested queues.

25.3 Configuring LANZ

LANZ is disabled by default and must be enabled to function. Upper and lower queue-length thresholds can be defined for individual interfaces.

These sections describe the basic LANZ configuration steps:

- Section 25.3.1: Enabling and Disabling LANZ
- Section 25.3.2: Setting LANZ Congestion Thresholds
- Section 25.3.3: Logging LANZ Congestion Events
- Section 25.3.4: Using LANZ Reports
- Section 25.3.5: Streaming LANZ Data

25.3.1 Enabling and Disabling LANZ

For the switch to collect and display latency information, LANZ must be enabled. The **queue-monitor length (global configuration mode)** command enables LANZ with the current settings, or with the default settings if none have been configured. LANZ is disabled by default.

When LANZ is enabled, the switch monitors queue lengths on all ports and queue length data is available in the following forms:

- syslog data (see **queue-monitor length log**)
- CLI display (see **show queue-monitor length**)
- CSV-format ouput (see **show queue-monitor length csv**)
- data stream (see **queue-monitor streaming**)

To disable LANZ globally, enter the **no queue-monitor length** command in global configuration mode. Disabling LANZ globally also discards LANZ log data, but retains settings. To disable LANZ on an individual interface, enter the **no queue-monitor length** command in interface ethernet configuration mode.

Examples

- This command enables LANZ on the switch.

```
switch(config)#queue-monitor length
```

- This command disables LANZ on the switch.

```
switch(config)#no queue-monitor length
```

- These commands disable LANZ on Ethernet interface 7.

```
switch(config)#interface ethernet 7
switch(config-if-Et7)#no queue-monitor length
```

25.3.2 Setting LANZ Congestion Thresholds

When LANZ is enabled on the switch, it generates over-threshold events when queue lengths on any monitored interface exceed the upper threshold value and continues generating them until all the queue lengths on that interface drop back below the lower threshold.

25.3.2.1 Congestion Thresholds on FM4000 Switches

Queue lengths are measured in 512-byte segments. The default threshold values are 512 segments and 256 segments. To change the threshold values for a specific interface, use the **queue-monitor length (interface configuration mode)** command.

Example

- These commands set the upper and lower queue-length thresholds on Ethernet interface 5 to 300 segments and 200 segments.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#queue-monitor length thresholds 300 200
switch(config-if-Et5)#
```

25.3.2.2 Congestion Thresholds on Petra Switches

Queue lengths are measured in bytes. The upper threshold value can be set to between 2 and 52428800 bytes (the default value is 52428800 bytes). To change the upper threshold value for a specific interface, use the **queue-monitor length (interface configuration mode)** command.

Example

- These commands set the upper queue-length threshold on Ethernet interface 5 to 2614400 bytes.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#queue-monitor length thresholds 2614400
switch(config-if-Et5)#
```

25.3.3 Logging LANZ Congestion Events

To generate syslog messages when queue lengths on an interface exceed its upper threshold, enable logging with the **queue-monitor length log** command. When logging is enabled, a log message is generated each time one or more queues on an interface exceed the upper threshold value for that interface (see **queue-monitor length (interface configuration mode)**). Once an interface is over threshold, additional messages are generated at a maximum rate of one per ***interval*** as long as the queue length remains above the lower threshold for that interface. No syslog message is generated when queue length drops back under threshold.

Queue length information is not included in log messages, but can be accessed by displaying LANZ data or exporting reports.

Examples

- This command enables queue-length over-threshold logging with a minimum interval of 10 seconds between messages for a given interface.

```
switch(config)#queue-monitor length log 10
```

- This command disables queue-length over-threshold logging on the switch.

```
switch(config)#queue-monitor length log 0
```

- This is an example of a queue-length log message.

```
Oct 27 12:48:22 switch QUEUE_MONITOR-6-LENGTH_OVER_THRESHOLD: Interface
Ethernet6 queue length is over threshold of 512, current length is 1024.
```

25.3.4 Using LANZ Reports

LANZ reports (LANZ data stored in the LANZ data buffer on the switch) can be displayed in the CLI or exported to a CSV file.

25.3.4.1 Viewing LANZ Reports in the CLI

When LANZ is enabled, the **show queue-monitor length** command displays a report of recent over-threshold events for a range of interfaces or for all interfaces. Output can be limited to a specified number of seconds or records; by default, the command displays data for all interfaces, limited to the last 1000 records. When LANZ data is displayed in the CLI, the most recent events are listed first.

Example

- This command displays the last 100 records for Ethernet interfaces 6 through 8.

```
switch#show queue-monitor length ethernet 6-8 limit 100
Report generated at 2010-01-01 12:56:13

Time                         Interface   Queue length (segments, 1 to 512 bytes)
-------------------------------------------------------------------------------
0:00:07.43393 ago            Et6         1049
0:00:39.22856 ago            Et7         2039
1 day, 4:33:23.12345 ago     Et6         1077
```

25.3.4.2 Saving LANZ Reports as CSV Files

When LANZ is enabled, the **show queue-monitor length csv** command creates a CSV report of the last 100,000 over-threshold events on the switch. When LANZ data is displayed as a report, the oldest events are listed first.

Example

- This command creates a CSV report of the last 100,000 over-threshold events and appends them to a file named dump.txt on the switch.

```
switch#show queue-monitor length csv >> file:/tmp/dump.txt
```

Report contents:

```
admin@switch head /tmp/dump.txt
Report generated at 2011-03-04 00:59:10
2010-01-01 12:56:13.45679,"Et7",2039
2010-01-01 12:56:34.12340,"Et6",1049
```

25.3.5 Streaming LANZ Data

To support analysis of latency conditions, the switch can be configured to stream LANZ congestion and configuration data. The switch streams LANZ data via TCP in Google protocol buffer format through port 50001 and through the management interface.

You must create a client application to receive the streaming data. By default, the switch will accept up to 10 client connections for streaming LANZ data. This limit can be configured up to a maximum of 100. Maximum connections can be configured when LANZ is disabled.

25.3.5.1 Enabling and Disabling LANZ Data Streaming

LANZ data streaming is disabled by default. To enable streaming, issue the **no** form of the **shutdown (queue-monitor-streaming configuration)** command in queue-monitor streaming configuration mode. To disable streaming, use the **shutdown (queue-monitor-streaming configuration)** command.

When streaming is disabled, a message is sent to any connected clients and the connections are closed.

To ensure client access to LANZ data, add a rule to any relevant ACL permitting traffic destined for the LANZ port (50001) before initiating a client connection for streaming from a remote host. A static rule (sequence number 130) in the default control plane ACL permits LANZ traffic, but a similar rule must be added to any user-created ACL.

Examples

- These commands enable the streaming of LANZ data from the switch.

```
switch(config)#queue-monitor streaming
switch(config-qm-streaming)#no shutdown
switch(config-qm-streaming)#
```

- These commands disable LANZ data streaming.

```
switch(config)#queue-monitor streaming
switch(config-qm-streaming)#shutdown
switch(config-qm-streaming)#
```

25.3.5.2 Configuring Maximum Connections

By default, the switch will accept a maximum of 10 client connections for LANZ data streaming. This maximum can be configured using the **max-connections** command. If a client connects to the switch after the limit has been reached, an error message is sent to the client and the connection is closed.

Example

- This command sets the maximum number of client connections the switch accepts for LANZ data streaming to 50.

```
switch(config-qm-streaming)#max-connections 50
```

25.3.5.3 LANZ Streaming Messages

When streaming is enabled, LANZ sends a message whenever a congestion event or a configuration event occurs. The messages are streamed in Google protocol buffer format.

Configuration Messages

A configuration message is sent whenever a change is made to the LANZ configuration settings on the switch. The switch also sends a configuration message when a new client connection is established.

The configuration message includes the following information:

- **timestamp** time of change in configuration in micro-seconds (UTC).
- **lanzVersion** LANZ feature version.
- **numOfPorts** number of ports in the switch.
- **segmentSize** segment size.
- **maxQueueSize** maximum queue size in segments.
- **intfName** name of the port.
- **switchId** ID of the chip on a multi-chip system.
- **portId** ID of the port.
- **internalPort** "true" if it is an internal port.
- **highThreshold** higher threshold value.
- **lowThreshold** lower threshold value.

Congestion Messages

A congestion message is sent whenever LANZ generates an over-threshold event.

The congestion message includes the following information:

- **timestamp** time of congestion in micro-seconds (UTC).
- **intfName** name of the port.
- **switchId** ID of the chip on a multi-chip system.
- **portId** ID of the port.
- **queueSize** queue size in segments at time of congestion.

25.3.5.4 Creating the LANZ Client

For a client device to receive streaming data from the LANZ server, it must be running a client application designed to receive LANZ data. Client programs must be based on the Google protocol buffer schema file describing the structure of the congestion and configuration messages which LANZ streams.

Google Protocol Buffers

Google protocol buffers provide an efficient mechanism for serializing LANZ data for streaming. A protocol buffer package is needed in order to run a LANZ client.

The latest version of the Google protocol buffer source code is available at this address: http://code.google.com/p/protobuf/downloads/list

LANZ Message Schema

LANZ client applications must be designed based on the LANZ protocol buffer schema, which defines the format and contents of the streamed messages. The schema file is shown below, and is also available on the Arista FTP site at this address: ftp://ftp.aristanetworks.com/data/ar/Lanz.proto

```
package LanzProtobuf;

message ConfigRecord {
    required uint64 timestamp = 1; // Time of change in configuration in micro-seconds (UTC)
    required uint32 lanzVersion = 2; // LANZ feature version
    required uint32 numOfPorts = 3; // Num of ports in the switch
    required uint32 segmentSize = 4; // Segement size
    required uint32 maxQueueSize = 5; // Maximum queue size in segments
    message PortConfigRecord {
        required string intfName = 1; // Name of the port
        required uint32 switchId = 2; // Id of the chip on a multi-chip system
        required uint32 portId = 3; // Id of the port
        required bool internalPort = 4; // 'True' if it's an internal port
        required uint32 highThreshold = 5; // Higher threshold
        required uint32 lowThreshold = 6; // Lower threshold
    }

    repeated PortConfigRecord portConfigRecord = 6; // Lanz config details of each port
}

message CongestionRecord {
    required uint64 timestamp = 1; // Time of congestion in micro-seconds (UTC)
    required string intfName = 2; // Name of the port
    required uint32 switchId = 3; // Id of the chip on a multi-chip system
    required uint32 portId = 4; // Id of the port
    required uint32 queueSize = 5; // Queue size in segments at time of congestion
}

message ErrorRecord {
    required uint64 timestamp = 1; // Time of event in micro-seconds (UTC)
    required string errorMessage = 2; // Text message
}

message LanzRecord {
    optional ConfigRecord configRecord = 1;
    optional CongestionRecord congestionRecord = 2;
    optional ErrorRecord errorRecord = 3;
}
```

Implementation Procedure

The following steps create and install a functional client to receive streamed LANZ data. This procedure assumes a functional Python programming environment.

1. On the device which is to receive the streamed LANZ data, download the protocol buffers source code from Google at this address: http://code.google.com/p/protobuf/downloads/list
2. Extract the source code.
3. Go to the "python" directory in the extracted package, and run **setup.py** to install the Python library.
4. Download the example client from the Arista FTP server at this address: ftp://ftp.aristanetworks.com/data/ar/lanz_client.py
5. Run lanz_client.py -h to activate the LANZ client.

25.4 LANZ Commands

LANZ Commands: Global Configuration

- queue-monitor length (global configuration mode) . . . . . Page 1155
- queue-monitor length log . . . . . Page 1158
- queue-monitor streaming . . . . . Page 1159

LANZ Commands: Interface Ethernet Configuration Mode

- queue-monitor length (interface configuration mode) . . . . . Page 1156

LANZ Commands: Queue-Monitor Streaming Configuration Mode

- max-connections . . . . . Page 1154
- shutdown (queue-monitor-streaming configuration) . . . . . Page 1163

LANZ Display Commands

- show queue-monitor length . . . . . Page 1160
- show queue-monitor length csv . . . . . Page 1161
- show queue-monitor length status . . . . . Page 1162

max-connections

The **max-connections** command sets the maximum number of client connections the switch accepts for streaming LANZ data. The default maximum is 10 connections. To stream LANZ data, you must use the **queue-monitor streaming** command to enable LANZ data streaming.

Command Mode

Queue-Monitor-Streaming Configuration

Command Syntax

max-connections *connections*

Parameters

- *connections* maximum number of simultaneous LANZ streaming client connections the switch will accept. Values range from 1 through 100.

Restrictions

This command is available only on FM4000 platform switches.

Examples

- This command sets the maximum number of client connections the switch accepts for LANZ data streaming to 50.

```
switch(config-qm-streaming)#max-connections 50
switch(config-qm-streaming)#
```

queue-monitor length (global configuration mode)

The **queue-monitor length (global configuration mode)** command enables LANZ with the current settings, or with the default settings if LANZ has not yet been configured. LANZ is disabled by default.

When LANZ is enabled, the switch monitors queue lengths on all ports and generates over-threshold events when an output queue becomes congested. Over-threshold event data is available in the following forms:

- syslog data (see **queue-monitor length log**)
- CLI display (see **show queue-monitor length**)
- CSV-format ouput (see **show queue-monitor length csv**)
- data stream (see **queue-monitor streaming**)

The **no queue-monitor length** and **default queue-monitor length** commands entered in global configuration mode disable LANZ and discard LANZ log data, but retain settings. LANZ settings include:

- logging settings (see **queue-monitor length log**)
- queue length thresholds (see **queue-monitor length (interface configuration mode)**)
- data streaming settings (see **queue-monitor streaming**)

Command Mode

Global Configuration

Command Syntax

```
queue-monitor length
no queue-monitor length
default queue-monitor length
```

Restrictions

This command is available on FM4000 and Petra platform switches.

Examples

- This command enables LANZ on the switch.

```
switch(config)#queue-monitor length
switch(config)#
```

- This command disables LANZ on the switch.

```
switch(config)#no queue-monitor length
switch(config)#
```

queue-monitor length (interface configuration mode)

The **queue-monitor length (interface ethernet configuration mode)** command sets queue length thresholds to define "congested" on the command-mode interface for purposes of LANZ reporting. If LANZ is enabled (see **queue-monitor length (global configuration mode)**), an over-threshold event is generated when one or more queues on the interface exceed the upper threshold, and over-threshold events continue to be generated until all queue lengths on the interface drop below the lower threshold. (To log these events, use the **queue-monitor length log** command.)

Entering the **no queue-monitor length** command in interface configuration mode disables LANZ on the interface. Entering either the **queue-monitor length (interface ethernet configuration mode)** command or the **default queue-monitor length** command enables LANZ on the interface by removing the **no queue-monitor length** command from the configuration.

The **no queue-monitor length** and **default queue-monitor length** commands in interface configuration mode both erase custom queue length threshold settings for the interface.

Command Mode

Interface Ethernet Configuration

Command Syntax

```
queue-monitor length LIMITS
```

Parameters

- *LIMITS* the threshold value or values (settings vary by platform). Values include
  - **thresholds** *upper_threshold* *lower_threshold* Upper and lower thresholds (FM4000 platform).
    - *upper_threshold* queue length in 512-byte segments that triggers an over-threshold event. Must be higher than *lower_threshold*. The minimum value is 2. The maximum is the largest number of segments which can be queued before packets are dropped, and varies based on factors including flow control state and private buffer settings. Default setting is 512.
    - *lower_threshold* lower queue length threshold in 512-byte segments. When logging is enabled, an over-threshold interface continues generating over-threshold events until all its queues drop back below this length. Must be lower than *upper_threshold*. Values range from 1 to 3188. Default setting is 256.
  - **threshold** *upper_threshold* Upper threshold only (Petra platforms).
    - *upper_threshold* is the queue length in bytes that triggers an over-threshold event. Values range from 2 to 52428800 bytes. Default setting is 52428800.

Guidelines

FM4000 platform switches: Queue length is measured in segments of 512 bytes. Default upper threshold is 512 segments and lower threshold is 256 segments. Both upper and lower thresholds are configurable.

On the Petra platform, queue length is measures in bytes. Only the upper threshold is configurable, and it is set at a default value of 52428800 bytes.

Examples

- These commands set the upper and lower queue-length thresholds on Ethernet interface 5 to 300 segments and 200 segments.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#queue-monitor length thresholds 300 200
```

- These commands reset the upper and lower queue-length thresholds on Ethernet interface 5 to their default values.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#default queue-monitor length thresholds
```

- These commands disable LANZ on Ethernet interface 5.

```
switch(config)#interface ethernet 5
switch(config-if-Et5)#no queue-monitor length
```

queue-monitor length log

The **queue-monitor length log** command enables logging of queue-length over-threshold events when LANZ is enabled on the switch (see **queue-monitor length (global configuration mode)**). When logging is enabled, a log message is generated each time one or more queues on an interface exceed the upper threshold value for that interface (see **queue-monitor length (interface configuration mode)**). Once an interface is over threshold, additional messages are generated at a maximum rate of one per ***interval*** as long as the queue length remains above the lower threshold for that interface. No syslog message is generated when queue length drops back under threshold.

Logging is disabled by default.

Log messages do not include queue length information. To view queue length information, use the **show queue-monitor length** or **show queue-monitor length csv** command.

The **queue-monitor length log** command with an interval value of 0 disables event logging.

Command Mode

Global Configuration

Command Syntax

queue-monitor length log *interval*

Parameters

- *interval* – minimum interval in seconds between logged messages from a single interface.
  - **0** queue-length logging is disabled on the switch (the default setting).
  - **1 to 65535** minimum logging interval (in seconds).

Restrictions

This command is available on FM4000 and Petra platform switches.

Examples

- This command enables over-threshold logging with a minimum interval of 10 seconds between messages for a given interface.

```
switch(config)#queue-monitor length log 10
```

- This command disables queue-length over-threshold logging on the switch.

```
switch(config)#queue-monitor length log 0
```

- This is an example of a queue-length log message.

```
Oct 27 12:48:22 switch QUEUE_MONITOR-6-LENGTH_OVER_THRESHOLD: Interface
Ethernet6 queue length is over threshold of 512, current length is 1024.
```

queue-monitor streaming

The **queue-monitor streaming** command places the switch in queue-monitor-streaming configuration mode. Queue-monitor-streaming configuration mode is not a group change mode; ***running-config*** is changed immediately upon command entry. The exit command does not affect ***running-config***.

To enable LANZ data streaming on the switch, use the **no** form of the **shutdown (queue-monitor-streaming configuration)** command.

The **exit** command returns the switch to global configuration mode.

Command Mode

Global Configuration

Command Syntax

```
queue-monitor streaming
```

Restrictions

This command is available only on FM4000 platform switches.

Commands Available in queue-monitor streaming Configuration Mode

- **max-connections**
- **shutdown (queue-monitor-streaming configuration)**

Example

- This command places the switch in queue-monitor streaming configuration mode.

```
switch(config)#queue-monitor streaming
switch(config-qm-streaming)#
```

show queue-monitor length

The **show queue-monitor length** command displays a report of recent over-threshold events for a range of interfaces or for all interfaces. Output can be limited to a specified number of seconds or records. The most recent events are listed first. By default, the command displays data for all interfaces, limited to the last 1000 records. Newest events are listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays "queue-monitor is disabled."

Command Mode

EXEC

Command Syntax

show queue-monitor length [***INTERFACES***] [***LIMIT***]

Parameters

- *INTERFACES* interface type and number for report. Values include:
  - <no parameter> displays information for all interfaces.
  - **ethernet** *e-range* e-range formats include a number, number range, or comma-delimited list of numbers and ranges
- *LIMIT* optional limiting parameters for report. Values include:
  - <no parameter> displays the last 1000 records.
  - **limit** *number* **samples** displays the last *number* records. Values range from 1 to 1000000.
  - **limit** *number* **seconds** displays all records generated during the last *number* seconds. Values range from 1 to 1000000.

Example

- This command displays the last 100 records for Ethernet interfaces 6 through 8.

```
switch>show queue-monitor length ethernet 6-8 limit 100
Report generated at 2010-01-01 12:56:13

Time                        Interface   Queue length (segments, 1 to 512 bytes)
--------------------------------------------------------------------------------
0:00:07.43393 ago             Et6         1049
0:00:39.22856 ago             Et7         2039
1 day, 4:33:23.12345 ago      Et6         1077
```

show queue-monitor length csv

The **show queue-monitor length csv** command creates a CSV report of the last 100,000 over-threshold events on the switch. Oldest events are listed first.

LANZ must be enabled to use this command (see **queue-monitor length (global configuration mode)**). If LANZ is disabled, the command displays "queue-monitor is disabled."

Command Mode

EXEC

Command Syntax

```
show queue-monitor length csv [INTERFACES] [DESTINATION]
```

Parameters

- *INTERFACES* interface type and number for report. Values include:
  - — <no parameter> displays information for all interfaces.
  - — **ethernet** *e-range* e-range formats include a number, number range, or comma-delimited list of numbers and ranges
- *DESTINATION* where the report data is sent. Values include:
  - — <no parameter> displays report in the CLI.
  - — **>** *url* exports report to the file at the specified URL, overwriting the file if it exists.
  - — **>>** *url* appends the report data to the file at the specified URL.

Guidelines

In EXEC mode, the command sends the report to the CLI. In Privileged EXEC mode, the DESTINATION parameter is used to send the report to a specified file.

Example

- This command creates a CSV report of the last 1000 over-threshold events and appends them to a file named dump.txt on the switch.

```
switch#show queue-monitor length csv >> file:/tmp/dump.txt
```

Report contents:

```
admin@switch head /tmp/dump.txt
Report generated at 2011-03-04 00:59:10
2010-01-01 12:56:13.45679,"Et7",2039
2010-01-01 12:56:34.12340,"Et6",1049
```

show queue-monitor length status

The **show queue-monitor length status** command displays the current LANZ configuration for the switch and for each interface.

Command Mode

EXEC

Command Syntax

show queue-monitor length status

Examples

- This command displays the current LANZ configuration. In this example, custom thresholds have been set on Ethernet interface 1 and LANZ has been disabled on Ethernet interface 15.

```
switch(config)#show queue-monitor length status
queue-monitor length disabled
Segment size in bytes            :   512
Maximum queue length in segments :  3188
Syslog interval in seconds       :    10
Port thresholds in segments:
 Port High threshold  Low threshold
  Et1             40              5
  Et2            512            256
  Et3            512            256
  Et4            512            256
  Et5            512            256
  Et6            512            256
  Et7            512            256
  Et8            512            256
  Et9            512            256
 Et10            512            256
 Et11            512            256
 Et12            512            256
 Et13            512            256
 Et14            512            256
 Et15           disabled
 Et16            512            256
 Et17            512            256
 Et18            512            256
 Et19            512            256
 Et20            512            256
 Et21            512            256
 Et22            512            256
 Et23            512            256
 Et24            512            256
```

shutdown (queue-monitor-streaming configuration)

The **shutdown** command disables the streaming of LANZ data to external clients. The **no shutdown** command enables LANZ data streaming. Streaming is disabled by default.

Command Mode

Queue-Monitor-Streaming Configuration

Command Syntax

```
shutdown
no shutdown
```

Restrictions

This command is available only on FM4000 platform switches.

Example

- These commands enable the streaming of LANZ data on the switch.

```
switch(config)#queue-monitor streaming
switch(config-qm-streaming)#no shutdown
switch(config-qm-streaming)#
```

Chapter 26

VM Tracer

This chapter describes VM Tracer configuration and usage and contains these sections:

- Section 26.1: VM Tracer Introduction
- Section 26.2: VM Tracer Conceptual Overview
- Section 26.3: VM Tracer Configuration Procedures
- Section 26.4: VM Tracer Configuration Commands

26.1 VM Tracer Introduction

VM Tracer is a switch feature that determines the network configuration and requirements of connected VMWare hypervisors. The switch uses VMWare's SOAP XML API to discover VMWare host server components, including

- instantiated VMs with their network configuration (VLANs and distributed/virtualSwitches).
- server hardware IPMI data which can be shown to the network manager.

VM Tracer also supports adaptive auto-segmentation, which automatically provisions and prunes VLANs from server-switched ports as VMs are instantiated and moved within the data center.

26.2 VM Tracer Conceptual Overview

Cloud operating systems manage large virtualized computing infrastructures, including software and hardware. Cloud operating systems consist of virtual machines and hypervisors:

- A virtual machine (VM) is a software implementation of a computer that operates as running on dedicated physical hardware. Multiple VMs share physical machine resources from a single physical device. Each VM is controlled by its operating system.
- A hypervisor, also called a virtual Machine Manager (VMM), is software that manages multiple operating systems running concurrently on a physical device.

VM Tracer tracks activity of VMs that are controlled by hypervisors connected to the switch's Ethernet or LAG ports. VM Tracer supports vSphere 4.x – VMware's cloud operating system. vSphere version 4.x features include dynamic virtual switches (vdswitches) and VM movement among VMWare servers (VMotion).

vSphere 4.x components include:

- ESX and ESXi: hypervisors that run on VMWare host server hardware.
- vCenter Server: centralized tool that manages multiple servers running VMware hypervisors.

vCenter manages ESX hosts and VMs through a central database. VM Tracer identifies interfaces connected to a specified ESX host and sends discovery packets on interfaces where VM Tracer is enabled. The ESX host updates the vCenter when it receives a discovery packet. VM Tracer reads this data from the vCenter to associate the ESX host to the connected switch ports.

VM Tracer connects to a maximum of four vCenters through a SOAP (Simple Object Access Protocol) API to discover VMs in the data centers that the vCenters manage. VM Tracer maintains a list of VMs in the data center and gathers network related information about each VM, including the number of Vnics (virtual network interface card), the MAC address of each Vnic, the switch to which it connects, and the host on which it resides. VM Tracer also identifies the host nics connected to the switch through the bridge MAC address and the interface port name. VM Tracer then searches for VMs on this host and connected to the vswitch or dvswitch whose uplink is mapped to the connected nic.

For each connected interface, VM Tracer creates a VM Table that lists its active VMs, sorted by Vnic MAC address. Each VM entry includes its name, Vnic name, VLAN, switch name, datacenter name, and portgroup. An entry is deleted when the corresponding VM is removed, moved to a different host, or its Vnic is no longer part of the vswitch or dvswitch. An entry is added when a VM is created or moved to a host connected to the interface. VM Tracer monitors vCenter for VM management updates. If an interface goes down, all VM entries for that interface are removed from the VMTable.

26.3 VM Tracer Configuration Procedures

The following sections describe the session configuration process and the procedure for enabling VM tracer on individual interfaces. The switch defines vmtracer configuration mode and VMtracer mode:

- vmtracer configuration mode is a command mode for configuring VM Tracer monitoring sessions.
- VMtracer mode is defines an interface state where discovery packets are sent to attached vSwitches.

26.3.1 Configuring vCenter Monitoring Sessions

A VM Tracer session connects the switch to a vCenter server for downloading data about VMs and vSwitches managed by ESX hosts connected to the switch's ports. The switch supports four VM Tracer sessions.

The switch is placed in vmtracer configuration mode to edit session parameters, including the vCenter location and dynamic VLAN usage. Changes take effect by exiting vmtracer mode.

The **vmtracer session** command places the switch in vmtracer configuration mode for a specified session. The command either creates a new session or loads an existing session for editing.

Example

This command enters vmtracer configuration mode for the system_1 session.

```
switch(config)#vmtracer session system_1
switch(vmtracer-system_1)#
```

In vmtracer configuration mode, the **url**, **username (vmtracer mode)**, and **password (vmtracer mode)** commands specify the vCenter server's location and the account information that authenticates the switch to the vCenter. The url parameter must reference a fully formed secure url, such as **https://vcenter.democorp.com/sdk**.

Example

These commands specify the vCenter's url along with the username and password that allow the switch to access the vCenter.

```
switch(vmtracer-system_1)#url https://vcenterserver.company1.org/sdk
switch(vmtracer-system_1)#username a-switch_01
switch(vmtracer-system_1)#password abcde
```

Default session settings allow auto-segmentation, or the dynamic allocation and pruning of VLANs when a VM managed by the ESX host connected to the switch is created, deleted, or moved to a different host. The **autovlan disable** command prevents auto-segmentation, regardless of VM activity. The **allowed-vlan** command specifies the VLANs that may be added when a VM is added or moved. By default, all VLANs are allowed.

Example

This command disables auto-segmentation.

```
switch(vmtracer-system_1)#autovlan disable
```

Example

These commands enable auto-segmentation and limit the list of allowed VLANs to VLAN 1-2000.

```
switch(vmtracer-system_1)#no autovlan disable
switch(vmtracer-system_1)#allow-vlan 1-2000
```

The **exit (vmtracer mode)** command returns the switch to Global Configuration mode and enables the VM Tracer session. Vmtracer configuration mode can be re-entered for this session to edit session parameters.

Example

This command exits vmtracer configuration mode.

```
switch(vmtracer-system_1)#exit
switch(config)#
```

The **no vmtracer session** command disables the session and removes it from ***running-config***.

Example

This command disables and deletes the system_1 VM Tracer session.

```
switch(config)#no vmtracer session system_1
```

26.3.2 Enabling vmtracer Mode

VMtracer mode is an interface setting that enables interfaces to send discovery packets to the connected vSwitch. The **vmtracer** command enables VMtracer mode on the configuration mode interface.

Example

These commands enable VMtracer mode on Ethernet 3 interface.

```
switch(config)#interface Ethernet3
switch(config-if-Et3)#vmtracer vmware-esx
```

The **no vmtracer** command disables vmtracer mode on the configuration mode interface.

Example

This command disables vmtracer mode on Ethernet 3 interface.

```
switch(config-if-Et3)#no vmtracer vmware-esx
```

26.3.3 Displaying VM Tracer Data

26.3.3.1 Displaying Session Status

The **show vmtracer session** command displays information about the specified session.

- without the ***detail*** parameter, the command displays connection parameters and status for the vCenter associated to the specified session.

  Example

  This command displays connection parameters for the vCenter associated with the system_1 session.

```
switch#show vmtracer session system_1
   vCenter URL https://vmware-vcenter1/sdk
   username arista
   password arista
   Session Status Disconnected
```

- with the ***detail*** parameter, the command displays connection status and data concerning messages the vCenter previously received from ESX hosts connected to the switch.

  Example

  This command displays connection parameters and message details for the vCenter associated with the system_1 session.

```
switch#show vmtracer session system_1 detail
   vCenter URL https://vmware-vcenter1/sdk
   username arista
   sessionState Connected
   lastStateChange 19 days, 23:03:59 ago
   lastMsgSent CheckForUpdatesMsg
   timeOfLastMsg 19 days, 23:14:09 ago
   resonseTimeForLastMsg 0.0
   numSuccessfulMsg 43183
   lastSuccessfulMsg CheckForUpdatesMsg
   lastSuccessfulMsgTime 19 days, 23:14:19 ago
   numFailedMsg 1076
   lastFailedMsg CheckForUpdatesMsg
   lastFailedMsgTime 19 days, 23:14:09 ago
   lastErrorCode Error -1 fault: SOAP-ENV:Client [no subcode]
   "End of file or no input: Operation interrupted or timed out after 600s
   send or 600s receive delay"
   Detail: [no detail]
   CheckForUpdates:
```

26.3.3.2 Displaying VM Interfaces

The **show vmtracer interface** command displays the VM interfaces (Vnics) that are active on switch interfaces where vmtracer mode is enabled. For each Vnic, the command displays the name of the attached VM, the adapter name, its VLAN, the VM power state, and the presence status of its MAC address in the switch's MAC table.

Example

This command displays the Vnics connected to all VM Tracer-enabled interfaces.

```
switch#show vmtracer interface

Ethernet8 : esx3.aristanetworks.com/vSwitch0/vmnic2
   VM Name VM Adapter VLAN Status
   esx3.aristanetworks.com vmk0 0 Up/Down
   vspheremanagement Network adapter 1 0 Up/Down

Ethernet15 : esx2.aristanetworks.com/vds/dvUplink1
   VM Name VM Adapter VLAN Status
   Openview Network adapter 1 123 Up/Down
   VmTracerVm Network adapter 1 123 Down/Down

Ethernet23 : esx3.aristanetworks.com/vds/dvUplink1
   VM Name VM Adapter VLAN Status

Ethernet24 : esx2.aristanetworks.com/None/None
   VM Name VM Adapter VLAN Status
```

26.3.3.3 Displaying VMs

The **show vmtracer vm** command displays VM interfaces (Vnics) accessible to the VM Tracer-enabled interfaces. For each active listed VM, the command displays its name, adapter, and the connected hypervisor.

Example

This command displays the VMs connected to all VM Tracer-enabled interfaces.

```
switch#show vmtracer vm
   VM Name VM Adapter Interface VLAN
   Openview Network adapter 1 Et15 123
   vspheremanagement Network adapter 1 Et8 0
   VmTracerVm Network adapter 1 Et15 123
   esx3.aristanetworks.com vmk0 Et8 0
```

Example

This command displays connection data for the VMs connected to all VM Tracer-enabled interfaces.

```
switch#show vmtracer vm detail
   VM Name Openview
      intf : Et15
      vnic : Network adapter 1
      mac : 00:0c:29:ae:7e:90
      portgroup : dvPortGroup
      vlan : 123
      switch : vds
      host : esx2.aristanetworks.com
```

26.4 VM Tracer Configuration Commands

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Commands

- vmtracer session . . . . . Page 1181

Interface Configuration (Ethernet and Port Channel) Commands

- vmtracer . . . . . Page 1180

VM Tracer Configuration Commands

- allowed-vlan . . . . . Page 1171
- autovlan disable . . . . . Page 1172
- exit (vmtracer mode) . . . . . Page 1173
- password (vmtracer mode) . . . . . Page 1174
- url . . . . . Page 1178
- username (vmtracer mode) . . . . . Page 1179

VM Tracer Display Commands

- show vmtracer interface . . . . . Page 1175
- show vmtracer session . . . . . Page 1176
- show vmtracer vm . . . . . Page 1177

allowed-vlan

The **allowed-vlan** command specifies the VLANs that may be added when a VM is added or moved from the hypervisor connected to the session specified by the vmtracer mode. By default, all VLANs are allowed.

Command Mode

vmtracer

Command Syntax

```
allowed-vlan VLAN_LIST
no allowed-vlan vlan
default allowed-vlan vlan
```

Parameters

- ***VLAN_LIST*** The VLAN list or the edit actions to the current VLAN list. Valid *v_range* formats include number, or number range.
  - *v_range* The list consists of the *v_range* VLANs.
  - **add** *v_range* The *v_range* VLANs are added to the current VLAN list.
  - **all** The list consists of all VLANs (1-4094).
  - **except** *v_range* The list consists of all VLANs except for those specified by *v_range*.
  - **none** The list of VLANs is empty.
  - **remove** *v_range* The *v_range* VLANs are removed from the current VLAN list.

Examples

- This command sets the list of allowed VLANs to 1 through 2000.

```
switch(vmtracer-system_1)#allow-vlan 1-2000
switch(vmtracer-system_1)#
```

- This command adds VLANs to 2501 through 3000.

```
switch(vmtracer-system_1)#allow-vlan add 2051-3000
switch(vmtracer-system_1)#
```

autovlan disable

Default VM Tracer session settings enable auto provisioning, which allows the dynamic assignment and pruning of VLANs when a VM attached to the ESX connected to the switch is created, deleted, or moved to a different ESX host. The autovlan setting controls auto provisioning.

The **autovlan disable** command disables auto provisioning, which prevents the creation or deletion of VLANs regardless of VM activity. The **allowed-vlan** command specifies the VLANs that may be added when a VM is added or moved. By default, all VLANs are allowed.

The **no autovlan disable** command enables the creation and deletion of VLANs caused by VM activity. This is the default setting.

Command Mode

vmtracer

Command Syntax

```
autovlan disable
no autovlan disable
default autovlan disable
```

Example

- This command disables dynamic VLAN creation or pruning within the configuration mode VM Tracer session.

```
switch(vmtracer-system_1)#autovlan disable
switch(vmtracer-system_1)#
```

exit (vmtracer mode)

The **exit (vmtracer mode)** command returns the switch to Global Configuration mode and enables the VM Tracer session. Changes to the VM Tracer session that were made in vmtracer mode are stored when the mode is exited.

Command Mode

vmtracer

Command Syntax

exit

Examples

- This command exits VM tracer mode.

```
switch(vmtracer-system_1)#exit
switch(config)#
```

password (vmtracer mode)

The **password** command specifies the token that authorizes the username to the vCenter associated with the VM Tracer mode session.

Command Mode

vmtracer

Command Syntax

password [***ENCRYPTTION***] [*password*]

Parameters

- ***ENCRYPTION*** encryption level of the password.
  - — <no parameter> *password* is a clear text string.
  - — **0** the *password* is a clear text string. Equivalent to <no parameter>.
  - — **7** the *password* is an encrypted string.
- *password* text that authenticates the username.
  - — *password* is a clear text string if ***ENCRYPTION*** specifies clear text
  - — *password* is an encrypted string if ***ENCRYPTION*** specifies an encrypted string.

Examples

This command configures ***1234*** as the clear text string that authorizes the username ***a-switch_01*** to the vCenter located at ***vcenterserver.company1.org***.

```
switch(vmtracer-system_1)#url https://vcenterserver.company1.org/sdk
switch(vmtracer-system_1)#username a-switch_01
switch(vmtracer-system_1)#password abcde
```

show vmtracer interface

The **show vmtracer interface** command displays the VM interfaces (Vnics) that are active on the VM Tracer enabled interface. For each Vnic, the command displays the name of the attached VM, the adapter name, its VLAN, the VM power state, and the presence status of its MAC address in the switch's MAC table.

Command Mode
Privileged EXEC

Command Syntax
show vmtracer interface [***INT_NAME***]

Parameters

- *INT_NAME* the interfaces to be configured. Values include:
  - <no parameter> Command displays data for all VM Tracer enabled interfaces.
  - **ethernet** *e_range* Ethernet interface range. Valid *e_range* formats include a number, number range, or comma-delimited list of numbers and ranges.
  - **port-channel** *p_range* Port Channel interface range. Valid *p_range* formats include a number, number range, or comma-delimited list of numbers and ranges.

Examples

- This command displays the Vnics connected to all VM Tracer enabled interfaces.

```
switch#show vmtracer interface

Ethernet8 : esx3.aristanetworks.com/vSwitch0/vmnic2
   VM Name                     VM Adapter          VLAN          Status
   esx3.aristanetworks.com     vmk0                0             Up/Down
   vspheremanagement           Network adapter 1   0             Up/Down

Ethernet15 : esx2.aristanetworks.com/vds/dvUplink1
   VM Name                     VM Adapter          VLAN          Status
   Openview                    Network adapter 1   123           Up/Down
   VmTracerVm                  Network adapter 1   123           Down/Down

Ethernet23 : esx3.aristanetworks.com/vds/dvUplink1
   VM Name                     VM Adapter          VLAN          Status

Ethernet24 : esx2.aristanetworks.com/None/None
   VM Name                     VM Adapter          VLAN          Status
```

- This command displays the Vnics connected to Ethernet 8 interface.

```
switch>show vmtracer interface Ethernet8

Ethernet8 : esx3.aristanetworks.com/vSwitch0/vmnic2
   VM Name                     VM Adapter          VLAN          Status
   esx3.aristanetworks.com     vmk0                0             Up/Down
   vspheremanagement           Network adapter 1   0             Up/Down
```

show vmtracer session

The **show vmtracer session** command displays information about a specified VM Tracer session.

Command Mode
Privileged EXEC

Command Syntax
show vmtracer session [***SESSION_LIST***] [***INFO_LEVEL***]

Parameters

- ***SESSION_LIST*** VM Tracer sessions for which the command returns information.
  - <no parameter> all configured VM Tracers sessions.
  - *session_name* name of one VM Tracer session.
- ***INFO_LEVEL*** specifies information that the command returns.
  - <no parameter> command displays connection parameters and status for the vCenter associated to the specified sessions.
  - **detail** command displays connection status and data concerning messages the vCenter previously received from ESX hosts connected to the switch.

Examples

- This command displays connection parameters for the vCenter associated to the system_1 session.

```
switch#show vmtracer session system_1
   vCenter URL                          https://vmware-vcenter1/sdk
   username                             arista
   password                             arista
   Session Status                       Disconnected
```

- This command displays connection parameters and message details from the vCenter associated to the system_1 session.

```
switch#show vmtracer session system_1 detail
   vCenter URL             https://vmware-vcenter1/sdk
   username                arista
   sessionState            Connected
   lastStateChange         19 days, 23:03:59 ago
   lastMsgSent             CheckForUpdatesMsg
   timeOfLastMsg           19 days, 23:14:09 ago
   resonseTimeForLastMsg   0.0
   numSuccessfulMsg        43183
   lastSuccessfulMsg       CheckForUpdatesMsg
   lastSuccessfulMsgTime   19 days, 23:14:19 ago
   numFailedMsg            1076
   lastFailedMsg           CheckForUpdatesMsg
   lastFailedMsgTime       19 days, 23:14:09 ago
   lastErrorCode           Error -1 fault: SOAP-ENV:Client [no subcode]
   "End of file or no input: Operation interrupted or timed out after 600s
   send or 600s receive delay"
   Detail: [no detail]
   CheckForUpdates:
```

show vmtracer vm

The **show vmtracer vm** command displays VMs interfaces (Vnics) that are accessible to VM Tracer enabled interfaces. For each active VM, the command displays the name of the VM, its adapter, and the hypervisor to which it connects.

Command Mode

Privileged EXEC

Command Syntax

show vmtracer vm [***INFO_LEVEL***] [***VM_LIST***]

Parameters

- *INFO_LEVEL* Specifies the information that the command returns.
  - — <no parameter> command displays connection parameters and status for the vCenter associated to the specified sessions.
  - — **detail** command displays connection status and data concerning messages the vCenter previously received from ESX hosts that received discovery packets from the switch.
- *VM_LIST* The virtual machines for which the command displays information. Options include:
  - — <no parameter> command returns information for all present VMs.
  - — *vm_name* command returns information only for specified VM.

Examples

- This command displays the VMs connected to all VM Tracer enabled interfaces.

```
switch#show vmtracer vm
   VM Name                    VM Adapter           Interface VLAN
   Openview                   Network adapter 1    Et15      123
   vspheremanagement          Network adapter 1    Et8       0
   VmTracerVm                 Network adapter 1    Et15      123
   esx3.aristanetworks.com    vmk0                 Et8       0
```

- This command displays connection data for the VMs connected to all VM Tracer enabled interfaces.

```
switch#show vmtracer vm detail
   VM Name   Openview
      intf       :        Et15
      vnic       :        Network adapter 1
      mac        :        00:0c:29:ae:7e:90
      portgroup  :        dvPortGroup
      vlan       :        123
      switch     :        vds
      host       :        esx2.aristanetworks.com
```

url

The **url** command specifies the vCenter server location that is monitored by the session being edited by the current vmtracer mode. The command must reference a fully formed secure url.

Command Mode

vmtracer

Command Syntax

url *url_name*

Parameters

- *url_name* location of the vCenter server. Valid formats include IP address (dotted decimal notation) and fully qualified domain name.

Example

- This command specifies the location of the vCenter monitored by the system_1 VM Tracer session.

```
switch(vmtracer-system_1)#url https://vcenterserver.company1.org/sdk
switch(vmtracer-system_1)#
```

username (vmtracer mode)

The **username** command identifies the switch's account name on the vCenter server. The switch uses this user name to access vCenter information.

Command Mode

vmtracer

Command Syntax

username *name_string*

Parameters

- *name_string* vCenter account user name. Parameter must match the user name configured on the vCenter.

Example

This command configures the user name for the vCenter associated with the system_1 session. The session uses this user name to log into the vCenter server.

```
switch(vmtracer-system_1)#username a-switch_01
switch(vmtracer-system_1)#
```

vmtracer

The **vmtracer** command enables vmtracer mode on the configuration mode interface. Interfaces with vmtracer mode enabled send discovery packets to the connected vSwitch.

The **no vmtracer** and **default vmtracer** commands disable vmtracer mode on the configuration mode interface by removing the corresponding **vmtracer** command from ***running-config***.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
vmtracer HOST_TYPE
no vmtracer HOST_TYPE
default vmtracer HOST_TYPE
```

Parameters

- ***HOST_TYPE*** the type of hypervisor that controls the vSwitch to which the interface connects.
  - **vmware-esx** ESX or ESXI hypervisor (VMware).

Examples

- These commands enable vmtracer mode on Ethernet 3 interface.

```
switch(config)#interface Ethernet 3
switch(config-if-Et3)#vmtracer vmware-esx
switch(config-if-Et3)#
This command disables vmtracer mode on Ethernet 3 interface.
switch(config-if-Et3)#no vmtracer vmware-esx
switch(config-if-Et3)#
```

vmtracer session

The **vmtracer session** command places the switch in vmtracer mode for the specified session. The command creates a new session or loads an existing session for editing.

A VM Tracer session connects the switch to a vCenter server at a specified location, then download data about VMs and vSwitches managed by ESX hosts connected to switch ports. The switch supports a maximum of four VM Tracer sessions.

VM Tracer session parameters are configured in vmtracer mode. Parameters configured in vmtracer mode include the vCenter location and dynamic VLAN usage.

VM Tracer mode commands include:

- **allowed-vlan**
- **autovlan disable**
- **exit (vmtracer mode)**
- **password (vmtracer mode)**
- **url**
- **username (vmtracer mode)**

The **no vmtracer session** and **default vmtracer session** commands disable the session and remove its configuration from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
vmtracer session name
no vmtracer session name
default vmtracer session name
```

Parameters

- *name* The label assigned to the VM Tracer session.

Examples

- This command enters vmtracer mode for the system_1 session.

```
switch(config)#vmtracer session system_1
switch(vmtracer-system_1)#
```

- This command disables the system_1 VM Tracer session. The system_1 session and all of its parameters are removed from ***running-config***.

```
switch(config)#no vmtracer session system_1
switch(config)#
```

Chapter 27

sFlow

This chapter describes Arista's implementation of sFlow, including configuration instructions and command descriptions. Topics covered by this chapter include:

- Section 27.1: sFlow Conceptual Overview
- Section 27.2: sFlow Configuration Procedures
- Section 27.3: SFlow Configuration Commands

27.1 sFlow Conceptual Overview

27.1.1 sFlow Technology

sFlow is a multi-vendor sampling technology that continuously monitors application level traffic flow at wire speed simultaneously on all interfaces. sFlow provides gigabit speed quantitative traffic measurements without impacting network performance.

sFlow.org is an international, multi-vendor, end-user forum that promotes sFlow sampling technology for monitoring and managing traffic in complex networks to support sFlow adoption by end users, network equipment vendors, and software application developers. sFlow.org web site is the authoritative source for information, specifications, developments, and products. The sFlow specification is published as RFC 3176. Source code for the sFlow agent and basic traffic analysis tools are freely available.

sFlow has the following network traffic monitoring characteristics:

- sFlow provides a network view of active route usage that measures network traffic.
- sFlow is scalable to 10 Gb/s without impacting switch performance or the network load.
- sFlow is implemented on a wide range of devices, without requiring additional memory and CPU.
- sFlow is an industry standard.

An sFlow configuration consists of:

- sFlow agents, embedded on network equipment, that monitors traffic and generates data.
- sFlow collectors that receive and analyze sFlow data.

Arista switches include an sFlow agent that monitors ingress data through all Ethernet interfaces.

27.1.1.1 sFlow Agents

The sFlow agent is a software process that runs as part of the network management software within an Arista switch. It combines interface counters and flow samples into sFlow datagrams that are sent to an sFlow collector. Packets typically include flow samples and state information of the forwarding/routing table entries associated with each sample.

The sFlow Agent performs minimal processing when packaging data into datagrams. Immediate data forwarding minimizes agent memory and CPU requirements.

27.1.1.2 sFlow Collector

An sFlow collector is a server that runs software that analyzes and reports network traffic. Collectors receive flow samples and counter samples respectively as sFlow datagrams from an sFlow agents. Arista switches reference a collector's IP address and UDP port as a configurable setting through a CLI command. Arista switches do not include sFlow collector software.

27.1.1.3 sFlow Data

The sFlow Agent uses two forms of sampling: statistical packet-based sampling of switched flows and time-based sampling of network interface statistics.

- **Switched flow sampling**: A sample is taken by either copying the packet's header or extracting feature data from the packet.
- **Interface statistics sampling**: Counter sampling extracts statistics by periodically polling each data source on the device.

sFlow implements flow sampling and counter sampling as part of an integrated system. An sFlow datagram incorporates both sample types.

27.1.2 Arista sFlow Implementation

Arista switches provide a single sFlow agent instance that samples ingress traffic from all Ethernet and port channel interfaces. The switch provides two levels of settings for enabling sFlow:

- a global setting that enables packet sampling on the entire switch.
- interface settings that control sampling on individual interfaces when sFlow is globally enabled.

sFlow default settings include:

- global: sFlow is globally disabled.
- Ethernet and port channel interfaces: sFlow is enabled on all interfaces when it is globally ***enabled***.

The switch performs sFlow polling when sFlow is globally enabled. The CLI provides commands that globally disable sampling while counter polling remains enabled. Sample enabling, while the switch continues polling, is not controllable on individual interfaces.

The switch sends sFlow datagrams to the collector destination located at an IP location specified by a global configuration command. If the collector destination is not configured, the switch samples data strings without transmitting the resulting datagrams.

Although the CLI enforces the configured sampling rate limit, it may drop samples if it cannot handle the number of samples it receives over a specified period. Under normal operation, the maximum packet sample rate is one per 16384 packets. The CLI allows for higher sampling rates by using the ***dangerous*** keyword.

The following lists describe sFlow's sampling behavior relative to different packet types:

- Packets that are sampled:
  - Bridged frames (to switchports, cpu)
  - Routed packets (except ip options and mtu violations)
  - Flooded packets
  - Multicast packets
- Packets that are not sampled:
  - LACP frames
  - LLDP frames
  - STP BPDUs
  - IGMP packets
  - PAUSE frames
  - PIM_HELLO packets
  - CRC error frames
  - Packets dropped by ACLs or due to VLAN violations
  - Routed packets with ip options or mtu violations

27.1.3 Petra Platform sFlow Implementation

sFlow implementation on Petra platform switches differ from sFlow implementation on other platforms as follows:

- Petra platform ports configured for mirroring cannot support sFlow.

  Ports configured for sFlow and mirroring will ignore sFlow and continue mirroring operations. sFlow configuration commands remain in place and take effect when mirroring is disabled on the port.
- sFlow packets use mini-multicast buffers on Petra platform switches. sFlow packets use unicast packets on other Arista platform switches.

A mini-multicast buffer is a Petra platform data structure that supports sFlow. Buffer space is allocated for unicast, multicast, and mini-multicast buffers. When implementing sFlow, it is recommended that a more buffer space is allocated to mini-multicast buffers with the platform petra buffers command.

Example

- The following command allocates 64 k buffer space to mini-multicast buffers:

```
switch(config)#platform petraA buffers mini-multicast 65536
! Command will cause interfaces to flap (links will go down/up).
Proceed with command? [confirm]y
switch(config)#
```

  The default setting is 8192 (8 k). Executing this command disrupts traffic on all switch ports.

27.2 sFlow Configuration Procedures

Implementing sFlow on an Arista switch consists of configuring the following agent parameters:

1. Collector location address
2. Agent source address
3. Polling interval.
4. Sampling rate.

After configuring the sFlow agent, sampling is initiated by globally enabling sFlow on the switch.

Configuring the collector location

The **sflow destination** command specifies the IP address and UDP port of an sFlow collector. The switch supports multiple collectors.

Example

- This command configures the switch to send sFlow data to collectors at 10.42.15.12, port 6100 and 10.52.12.2 port 6343 (the default sFlow port).

```
switch(config)#sflow destination 10.42.15.12 6100
switch(config)#sflow destination 10.52.12.2
```

Configuring the agent source address

The **sflow source** command specifies the source address that the switch places in all sFlow datagrams that it sends to the collector. This address is normally set to an IP address configured on the switch.

Example

- This command configures 14.2.9.21 as the sFlow source address.

```
switch(config)#sflow source 14.2.9.21
```

The **sflow source-interface** command can be alternatively used to specifies the interface from which an IP address is derived that the switch places in all sFlow datagrams that it sends to the collector. This address is normally set to an IP address configured on the switch.

Example

- This command configures VLAN interface 25 as the sFlow source interface. The switch enters the IP address for VLAN 25 in the source field of sFlow datagrams.

```
switch(config)#sflow source 14.2.9.21
```

running-config cannot simultaneously contain **sflow source** and **sflow source-interface** commands.

Configuring the polling interval

The **sflow polling-interval** command specifies the interval for sending counter data to the sFlow collector. The default interval is two seconds.

Example

- This command configures the switch to send sFlow data every ten seconds.

```
switch(config)#sflow polling-interval 10
```

Configuring the sampling rate

The **sflow sample** command sets the packet sampling rate. A rate of 16384 corresponds to an average sample of one per 16,384 packets.

Example

- This command configures the sFlow sampling rate as 65536 (one per 65,536 packets).

```
switch(config)#sFlow sample 65536
```

Enabling sFlow

The **sflow run** command globally enables sFlow on the switch. The **sflow enable** command controls sFlow operation on Ethernet and port channel interfaces when sFlow is globally enabled. The **sflow enable** command has no effect when sFlow is globally disabled.

Example

- These commands enable sFlow on the switch, then disables sFlow on Ethernet interface 10.

```
switch(config)#sflow run
switch(config)#interface ethernet 10
switch(config-if-Et10)#no sflow enable
```

27.3 SFlow Configuration Commands

This section contains descriptions of the CLI commands that this chapter references.

Global Configuration Commands

- sflow destination . . . . . Page 1190
- sflow polling-interval. . . . . Page 1192
- sflow run . . . . . Page 1193
- sflow sample . . . . . Page 1194
- sflow source. . . . . Page 1195
- sflow source-interface . . . . . Page 1196

Interface Configuration Commands

- sflow enable. . . . . Page 1191

Privileged EXEC Command

- clear sflow counters . . . . . Page 1189

sFlow Display Commands

- show sflow. . . . . Page 1197
- show sflow interfaces. . . . . Page 1199

clear sflow counters

The **clear sflow counters** command resets the global sFlow statistics, which includes the number of samples and sample pool. The hardware trigger count is not reset.

The **show sflow** command displays global sFlow statistics.

Command Mode

Privileged EXEC

Command Syntax

```
clear sflow counters
```

Example

- This command resets the sFlow counters.

```
switch#clear sflow counters
switch#
```

sflow destination

The **sflow destination** command specifies an sFlow collector IP address and UDP port. The switch supports sFlow collector addresses through multiple sFlow destination commands in ***running-config.***

The **no sflow destination** and **default sflow destination** commands remove the specified sFlow collector IP address by deleting the corresponding **sflow destination** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
sflow destination dest_addr [UDP_PORT]
no sflow destination dest_addr [UDP_PORT]
default sflow destination dest_addr [UDP_PORT]
```

Parameters

- *dest_addr* sflow collector's IP address.
- ***UDP_PORT*** sFlow collector's data reception port\. Options include:
  - <No parameter> port number 6343 (default).
  - *port_num* port number. Values range from 0 to 65536.

Example

- This command configures the switch to send sFlow data to the collector located at 10.42.15.12; the collector receives the data through UDP port 6100.

```
switch(config)#sflow destination 10.42.15.12 6100
switch(config)#
```

sflow enable

The **sflow enable** command enables sFlow on the configuration mode interface when sFlow is globally enabled. By default, sFlow is enabled on individual interfaces when sFlow is globally enabled (**sflow run**). The **sflow enable** command is required only when ***running-config*** contains a **no sflow enable** statement for the specified interface.

The **no sflow enable** command disables sFlow on the configuration mode interface. When sFlow is globally disabled, this command persists in ***running-config*** but has no effect on switch operation.

The **default sflow enable** command removes the corresponding **no sflow enable** command from ***running-config,*** enabling sFlow capability on the interface.

Command Mode

Interface-Ethernet Configuration
Interface-Port-Channel Configuration

Command Syntax

```
sflow enable
no sflow enable
default sflow enable
```

Examples

- These commands enable sFlow on the switch and disable sFlow on Ethernet interface 12.

```
switch(config)#sflow run
switch(config)#interface ethernet 12
switch(config-if-Et12)#no sflow enable
switch(config-if-Et12)#
```

- This command removes the **no sflow enable** command for Ethernet interface 12 from ***running-config***, enabling sFlow on the interface whenever sFlow is globally enabled.

```
switch(config-if-Et12)#sflow enable
switch(config-if-Et12)#
```

sflow polling-interval

The **sflow polling-interval** command specifies the counter's polling interval. The switch uses this interval to schedule a port's counter data transmissions to the sFlow collector.

The default interval is two seconds.

The **no sflow polling-interval** and **default sflow polling-interval** commands revert the polling interval to the default of two seconds by removing the **sflow polling-interval** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
sflow polling-interval interval_period
no sflow polling-interval
default sflow polling-interval
```

Parameters

- *interval_period* polling interval (seconds). Values range from 0 to 3600 (60 minutes). Default is 2.

Example

- This command configures the switch to send sFlow counter data every ten seconds.

```
switch(config)#sflow polling-interval 10
switch(config)#
```

sflow run

The **sflow run** command globally enables sFlow on the switch. The default sFlow global setting is ***disabled***. sFlow cannot be enabled on individual interfaces when it is globally disabled.

The **sflow enable** interface configuration command controls sFlow operation on individual Ethernet and port channel interfaces when sFlow is globally enabled. When sFlow is enabled globally, sFlow is also enabled on all interfaces by default.

The **no sflow run** and **default sflow run** commands globally disable sFlow on the switch.

Command Mode

Global Configuration

Command Syntax

```
sflow run
no sflow run
default sflow run
```

Examples

- This command enables sFlow on the switch.

```
switch(config)#sflow run
switch(config)#
```

- This command globally disables sFlow.

```
switch(config)#no sflow run
switch(config)#
```

sflow sample

The **sflow sample** command sets the packet sampling rate. The packet sampling rate defines the average number of ingress packets that pass through an interface for every packet that is sampled. A rate of 16384 corresponds to an average sample of one per 16,384 packets.

The **no sflow sample** and **default sflow sample** commands reset the packet sampling rate to the default of 1,048,576 by removing the **sflow sample** command from the configuration.

Command Mode

Global Configuration

Command Syntax

```
sflow sample SAMPLE_RATE
no sflow sample
default sflow sample
```

Parameters

- *SAMPLE_RATE* size of the packet sample from which one packet is selected. Default sample size is 1048576 packets. Options include:
  - *restricted_rate* where *restricted_rate* is an integer between 16384 to 16777216.
  - **dangerous** *any_rate* where *any_rate* is an integer between 1 to 16777216.

Examples

- This command configures the sFlow sampling rate as 65536 (one per 65,536 packets).

```
switch(config)#sFlow sample 65536
switch(config)#
```

- This command configures the sFlow sampling rate as 256 (one per 256 packets).

```
switch(config)#sFlow sample dangerous 256
switch(config)#
```

sflow source

The **sflow source** command specifies the address that is listed as the source in all sFlow datagrams that the switch sends to the collector. The source address is normally set to an IP address configured on the switch. This command cannot be used if ***running-config*** contains an **sflow source-interface** command.

The **no sflow source** and **default sflow source** commands remove the **sflow source** command from *running-config*.

Command Mode

Global Configuration

Command Syntax

```
sflow source source_addr
no sflow source
default sflow source
```

Parameters

- *source_addr* source IP address (dotted decimal notation).

Example

- This command configures 14.2.9.21 as the sFlow source address.

```
switch(config)#sflow source 14.2.9.21
switch(config)#
```

sflow source-interface

The **sflow source-interface** command specifies the interface from which the sFlow source IP address is derived. The switch enters the interface's IP address as the source in sFlow datagrams that it sends to the collector. This command cannot be used if ***running-config*** contains an **sflow source** command.

The **no sflow source-interface** and **default sflow source-interface** commands remove the **sflow source-interface** command from ***running-config***.

Command Mode

Global Configuration

Command Syntax

```
sflow source-interface INT_NAME
no sflow source-interface
default sflow source-interface
```

Parameters

- ***INT_NAME*** Interface type and number. Options include:
  - <no parameter> resets counters for all interfaces.
  - **interface ethernet** *e_num* Ethernet interface specified by *e_num*.
  - **interface loopback** *l_num* Loopback interface specified by *l_num*.
  - **interface management** *m_num* Management interface specified by *m_num*.
  - **interface port-channel** *p_num* Port-Channel Interface specified by *p_num*.
  - **interface vlan** *v_num* VLAN interface specified by *v_num*.

Example

- This command configures the sFlow source address as the IP address assigned to the loopback interface.

```
switch(config)#sflow source-interface loopback 0
switch(config)#
```

show sflow

The **show sflow** command displays configured sFlow parameters, operational status, and statistics.

The **show sflow interfaces** command displays the interfaces where sFlow is enabled.

Command Mode

Privileged EXEC

Command Syntax

show sflow [***INFO_LEVEL***]

Parameters

- *INFO_LEVEL* Specifies the information that the command displays: Options include:
  - — <no parameter> displays base information
  - — **detail** displays base information plus hardware sampling status and number of discarded samples.

Examples

- This command displays the base sFlow information.

```
switch#show sflow
Warning: displaying counters that may be stale
sFlow Configuration
-------------------
Destination IP: 171.67.90.3
Destination Port: 6343 ( default )
Source IP: 0.0.0.0 ( default )
Sample Rate: 16384
Polling Interval (sec): 2.0 ( default )

Status
------
Running: Yes
Polling On: Yes ( default )
Sampling On: Yes ( default )
Send Datagrams: No ( default )
Hardware Sample Rate: 16384

Statistics
----------
Total Packets: 20334189
Number of Samples: 1201
Sample Pool: 19677184
Hardware Trigger: 1205
Number of Datagrams: 356
```

- This command displays the expanded sFlow information.

```
switch#show sflow detail
Warning: displaying counters that may be stale
sFlow Configuration
-------------------
Destination IP: 171.67.90.3
Destination Port: 6343 ( default )
Source IP: 0.0.0.0 ( default )
Sample Rate: 16384
Polling Interval (sec): 2.0 ( default )

Status
------
Running: Yes
Polling On: Yes ( default )
Sampling On: Yes ( default )
Send Datagrams: No ( default )
Hardware Sample Rate: 16384
Hardware Sampling On: No

Statistics
----------
Total Packets: 20334189
Number of Samples: 1201
Sample Pool: 19677184
Hardware Trigger: 1205
Number of Datagrams: 356
Number of Samples Discarded: 0
```

show sflow interfaces

The **show sflow interfaces** command displays the interfaces where sFlow is enabled.

The **show sflow** command displays configured sFlow parameters, operational status, and statistics.

Command Mode
Privileged EXEC

Command Syntax
show sflow interfaces

Examples

- This command displays the show sflow interface message when sFlow is globally disabled.

```
switch#show sflow interfaces
sFlow Interface (s):
--------------------
sFlow is not running
```

- This command displays the show sflow interface message when sFlow is globally enabled and enabled on all interfaces.

```
switch(config)#sflow run
switch(config)#show sflow interfaces
sFlow Interface (s):
--------------------
Ethernet1
Ethernet2
Ethernet3
Ethernet4
Ethernet5
Ethernet6
Ethernet7
Ethernet8
Ethernet9
Ethernet10
Ethernet11
Ethernet12
Ethernet13
Ethernet14
Ethernet15
Ethernet16
Ethernet17
Ethernet18
Ethernet19
Ethernet20
Ethernet21
Ethernet22
Ethernet23
Ethernet24
```

Glossary

802.1Q. a networking standard that allows multiple bridged networks to transparently share the same physical network link without information leakage between networks. IEEE 802.1Q is also known as VLAN Tagging,

Access Control List (ACL). a list of attributes that routers use to filter network traffic when forwarding or blocking packets.

Bash. a Unix software shell.

Autonomous system (AS). A set of routers under a single administration.

Border Gateway Protocol (BGP). an Internet routing protocol that maintains a table of IP networks (prefixes) that designate network reachability among autonomous systems.

Broadcast Storm. extreme amounts of broadcast traffic that can consume enough network resources to prevent the network from transporting normal traffic.

class of service. a 3 bit field within an frame header that specifies a priority value of between 0 and 7 that Quality of Service (QoS) disciplines use to differentiate traffic.

Control Plane. the router architecture component that is concerned with drawing the network map, or the routing table information that defines the processing of inbound packets.

Control Plane Policing. a service that limits the rate of CPU bound control plane traffic to protect the CPU from unnecessary or denial of service traffic and gives priority to important control plane and management traffic.

Data Center Bridging Exchange (DCBX). a discovery and capability exchange protocol that conveys configuration and attribute information between network devices to ensure consistent configuration across the network.

Dynamic Host Control Protocol (DHCP). is a network protocol that hosts use, as DHCP clients, to retrieve IP address assignments and other configuration information.

Extensible Operating System (EOS). the network operating system that provides the interface between Arista switch hardware and the software controlling the switch and managing the network.

Equal Cost Multi-Path Routing (ECMP). a routing strategy that balances traffic over multiple paths designated by routing metric calculations.

Forced Autonegotiation. the configuration of a port to limit the speed to which it negotiate.

In Service Software Update (ISSU). a feature that allows updates to router software without disrupting packet forwarding.

Jumbo Frame. frames with more than 1,500 bytes of payload.

Layer 2 Tunneling Protocol (L2TP). a tunneling protocol that supports virtual private networks (VPNs).

Link Aggregation Protocol (LAP). a protocol that combines multiple ports in parallel to increase the link speed beyond the limits of any single port or to increase the redundancy for higher availability.

Link Layer Discovery Protocol (LLDP). a Data Link Layer protocol that network devices use to advertising of their identity, capabilities, and interconnections on local area networks.

Local Authentication. a method of providing authentication and authorization services for users that does not require accessing a remote device.

MAC Security. a switch feature that limits the number of MAC addresses that may appear on a port to a user-specified limit – typically one or two addresses.

Multicast Services. the simultaneous delivery of information to a group of destinations where messages are delivered over each link of the network only once and data is copied only when the links to the multiple destinations split.

Multi-Chassis Link Aggregation Protocol (MLAG). a method of configuring ports belonging to two cooperating switches such that they appear, to external devices, as an ordinary link aggregation group.

Multiple Spanning Tree Protocol (MSTP). an extension of the Rapid Spanning Tree Protocol that accommodates multiple VLAN groups.

Open Shortest Path First Protocol (OSPF). a link-state routing protocol used by Internet Protocol (IP) networks to route packets solely within a single routing domain.

Per-VLAN Rapid Spanning Tree (PVRST). an extension of the Rapid Spanning Tree Protocol that deploys a spanning tree for each VLAN.

Port Mirroring. a facility that sends a copy of network packets seen on one switch port to a network monitoring connection on another switch port.

Priority Flow Control (PFC). a link level flow control mechanism that is independently controllable for each Class of Service (CoS).

Quality of Service (QoS). a resource reservation control mechanism that provides different priorities to different applications, users, or data flows to guarantee specific performance levels or attributes to a data flow.

Rapid Spanning Tree Protocol (RSTP). an extension of the Spanning Tree Protocol that provides for faster spanning tree convergence after a topology change.

Remote Authentication Dial-In Service (RADIUS). a networking protocol that provides centralized Authentication, Authorization, and Accounting (AAA) management for computers accessing a network service.

Secure Shell (SSH). a network protocol that facilitates data exchanges through a secure channel between two network devices.

Simple Network Management Protocol (SNMP). a UDP-based network protocol used to monitor network-attached devices for conditions that warrant administrative attention.

Spanning Tree Protocol. a link layer network protocol that ensures a loop-free topology for any bridged LAN.The protocol creates a spanning tree within a mesh network of connected layer-2 bridges (typically switches) and disables links that are not part of the spanning tree to leave a single active path between any two network nodes.

Static Routing. the assignment of fixed network addresses to routers and other network devices.

Storm Control. a feature where a switch intentionally ceases forwarding all broadcast traffic when inbound broadcast frames consume a designated threshold bandwidth.

tcpdump. a common packet analyzer that intercepts and displays TCP/IP and other packets transmitted or received over a network to which the computer is attached.

Terminal Access Conroller Access Control System Plus (TACACS+). a protocol that provides separate authentication, authorization and accounting services for routers, network access servers, and other network devices through one or more centralized servers.

traceroute. a network tool that displays the routes taken by packets across an IP network.

tunneling. a method of sending payload over incompatible or untrusted networks by encapsulating data with a delivery protocol supported by the network.

Virtual Local Area Network (VLAN). a group of switches and routers that communicate as if they are attached to the same broadcast domain, regardless of their physical location.

virtual private networks (VPN). a computer network that is layered on top of an underlying network. Data travelling through a VPN is encapsulated from underlying network traffic.

Virtual Router Redundancy Protocol (VRRP). a redundancy protocol that increases the availability of default gateway servicing hosts on the same subnet through the definition of a virtual router. Two or more physical routers are configured to stand for the virtual router, with one actively routing packets and the others on standby in case of failure.

Index

For a list of configuration commands, see the Command Reference, starting on page 9

Symbols

?, question mark 55

Numerics

10 Gigabit Ethernet 250
10/100/1000BASE-T 250
1000BASE 250
10GbE 250
40G ports 255, 256
7050Q-16, port configuration 257
802.1ad 31, 386
802.1Q 31, 385

A

AAA . . . . . *see* Authorization, Authentication, Accounting
Aboot, boot loader
    Aboot password, recovery 43
    Aboot shell 208–211
    description 66, 197
abort (group change configuration mode command) 62
access control list, ACL
    ACL configuration command mode 460
    assigning 469
    commands 481–482, 484–487, 489–495, 497–499, 501–502, 504–508, 510–515, 520, 523, 525, 527, 529
    configuration 460–468
    counting 459, 465, 466
    creating 460
    description 30, 456–459
    discarding changes 461
    displaying 465–468
    editing 462–464
    IPv6-ACL configuration command mode 460
    logging 459, 464
    MAC-ACL configuration command mode 460
    rule tracking 459
    rules 456–458
    saving 460
    standard ACLs 458
    std-ACL configuration command mode 460
    std-IPv6-ACL configuration command mode 460
access ports (VLANs) 389
accessory kit 36
ACL . . . . . *see* access control list
ACL configuration command mode 460
address families (BGP) 897
address family (BGP) 906
address format
    IPv4 681
    IPv6 728
Address Resolution Protocol, ARP 682, 687
address-mask 55
address-wildcard 56
adjacencies 765, 839
adjacency changes, logging
    OSPFv2 768
    OSPFv3 842
admin username 36
advertisement timer (VRRP) 534
advertising routes (BGP) 896
agent (sFlow) 1184
agent (SNMP)
    description 1105
    extending 1112
aggregation, route (BGP) 896
aging time (dynamic MAC address) 332
alternate ports (STP) 566
anycast-rp (PIM) 1005, 1076, 1079, 1088
area assignments
    OSPFv3 844
area border router, ABR
    OSPFv2 764
    OSPFv3 838
ARP . . . . . *see* Address Resolution Protocol
AS path access list (BGP) 894
authentication (VRRP) 534

authorization 105
Authorization, Authentication, Accounting, AAA
commands 109–150
configuration 96–106
description 95–96
autonegotiation
configuration 258
displaying 261
autonomous system boundary router, ASDB
OSPFv2 764
OSPFv3 838
autonomous system, AS
OSPFv2 764
OSPFv3 838

B

backbone area
OSPFv2 765
OSPFv3 839
backup ports (STP) 566
backup router (VRRP) 531
banner 155
bash shell 66, 208
BGP . . . . . *see* Border Gateway Protocol
blocking state (STP) 566
boot loader . . . . . .*see* Aboot
boot-config file 44, 198
BOOTP 29
bootstrap message, BSM 1004
Bootstrap Protocol 29
bootstrap router, BSR 1004
Border Gateway Protocol, BGP
address families 897
address family 906
advertising routes 896
aggregation, route 896
AS path access list 894
commands 904–974
communities 895
configuration 891–900
description 31, 889–890
examples 901–903
hold time 892
keepalive message/period 892
multiprotocol 890
neighbor activation 898
neighbors 891
network route advertising 899
next hop 934, 935
out delay 936
redistributing routes 953
route maps 894
route reflection 893
router-BGP configuration command mode 891
BPDU (STP) . . . . . *see* Bridge Protocol Data Unit
bridge assurance (STP) 577
Bridge Protocol Data Unit, BPDU (STP)
BPDU filter 579
BPDU guard 578
description 567
bridge timers (STP) 567, 577
bridging 324
BSM . . . . . .*see* bootstrap messages
BSR. . . . . *see* bootstrap router
BSR border 1005

C

cable, serial port 36
candidate-bootstrap router, C-BSR 1004
candidate-rendezvous point, C-RP 1005
C-BSR. . . . . *see* candidate-bootstrap router
channel group
commands 294
configuration 289
description 287
chassis ID (SNMP) 1110
child modes 61
CIDR notation 55
Class of Service, CoS
CoS rewrite 638
Ethernet frame field 637
see also Quality of Service
clauses (route map) 470
clear text 96
CLI . . . . . *see* command line interface
CLI scheduler 57
collector (sFlow) 1184
command line interface, CLI
accessing 53
CLI scheduler 57
commands 68–75
command list 55
command modes
ACL configuration mode 460
console-management mode 40
control-plane configuration mode 343
daemon configuration mode 76
description 60–62
EXEC mode 60
global configuration mode 60
group change configuration modes 62
interface configuration modes 60
interface-Ethernet configuration mode 254
interface-loopback configuration mode 336
interface-management configuration mode 254
interface-port channel configuration mode 290
interface-VLAN configuration mode 392
IPv6-ACL configuration mode 460
IPv6-pfx configuration mode 477
MAC-ACL configuration mode 460
mc-tx-queue configuration mode 655
MLAG configuration mode 430
Privileged EXEC mode 60
prompts 183
protocol specific modes 60
queue-monitor streaming configuration mode . . 1149
route-map configuration mode 472
router-BGP configuration mode 891
router-OSPF configuration mode 767
router-OSPF3 configuration mode 841
router-RIP configuration mode 976
server-group-radius configuration command mode 118
server-group-tacacs+ configuration command mode 119
SSH-management mode 40
std-ACL configuration mode 460

std-IPv6-ACL configuration mode 460
Telnet-management mode 40
tx-queue configuration mode 648, 679
uc-tx-queue configuration mode 680
vmtracer configuration command mode 1166
commands, truncating 54
communities (BGP) 895
community access control (SNMP) 1108
community VLANs 387
congestion (LANZ) 1145, 1147
congestion events (LANZ) 1148
console 40
console port 36, 53
console settings, factory default 218
console-management command mode 40
contact string (SNMP) 1111
context-active VRF 339
contributor routes (BGP) 896
control plane
control-plane configuration mode 343
description 326
forwarding mode 375
multicast 993
policing 30
control sequences, prompt 156
copy running-config (command) 63
CoS ..... *see* Class of Service
cost, path (STP) 573
counting (ACL) 459, 465, 466
C-RP..... *see* candidate-rendezvous point
cursor movement keystrokes 54
customer VLAN (q-in-q network) 386
cut through (control plane) 375
cut through (data plane) 326, 338

D

daemon configuration mode 76
Data Center Bridging Exchange, DCBX 31
data plane 326, 338
DCS-7050Q-16, port configuration 257
dead interval
OSPFv2 772
OSPFv3 845
deadtime (RADIUS) 102
debounce 269
default route
IPv4 684
IPv6 730
default route to gateway 39
designated bridge, DB (STP) 565
designated port, DP (STP) 565
designated router priority (PIM-SM) 1006
designated router, DR (PIM-SM) 996
DHCP relay 29, 682, 687
DHCP server (ZTP configuration) 204
Differentiated Service Code Point, DSCP 637
directory structure 67
disable, dis (command) 61
disabled state (STP) 566
domain ID (MLAG) 424, 431
domain name 151
Domain Name Server, DNS 152
Domain Name System 151
dot1q tunnel port 391
DSCP ..... *see* Differentiated Service Code Point
duplex 258
dynamic MAC address 328, 332
dynamic routing 325
dynamic RPs (PIM-SM) 997, 1004

E

EBGP ..... *see* external BGP
ECMP..... *see* equal cost multi-path routing
edge ports (STP) 566
enable password
description 97
recovery 41
encapsulation, multicast 996
encrypted strings 96
encryption key
RADIUS 101
TACACS+ 99
engine ID (SNMP) 1109
environment control
commands 239–247
description and configuration 233–238
EOS CLI 29
EOS image
incorrectly configured 210
restoration 43
show version 46
transferring 45
upgrade 45
equal cost multi-path routing, ECMP 31, 330, 684
Ethernet
10 Gigabit 250
10/100/1000BASE-T 250
gigabit 250
physical layer 251
standards 249–250
Ethernet interface
configuration 254
description 253
QSFP+ modules 255
Ethernet management port 29, 38, 53
event monitor
configuration 157–158
description 157
queries 159
EXEC command mode 60
expanded communities (BGP) 895
extended communities (BGP) 896
Extensible Operating System, EOS 53
Exterior Gateway Protocol, EGP 889
external BGP, EBGP 889
external neighbors (BGP) 891

F

factory default configuration 42
fallback (LACP) 288, 291
fan modules 233
fan status, viewing 237
fast dropping (multicast) 994
FAT file system 197

feature set
layer 2 features 30
layer 3 features 31
feature to switch model availability table 32
FIB ..... *see* forwarding information base
file system 197
flag configuration, router advertisement (IPv6) 733
flash drive 67
flow control
commands 263–264
description 259
FM4000 switch models 59
forwarding information base, FIB
IPv4 684
IPv6 734
forwarding mode (control plane) 375
forwarding state (STP) 566
forwarding, hardware dependent (multicast) 994
forward-time (STP) 577
forward-time bridge timer (STP) 567
FQDN .....*see* fully qualified domain name
FTP 45
fullrecover (command) 43
fully meshed network topology 893
fully qualified domain name, FQDN 151

G

gateway
default route 39
displaying (IPv4) 685
GbE 250
general query message (IGMP) 995
gigabit Ethernet 250
global configuration command mode 60
global parameters
RADIUS 100
TACACS+ 99
Google protocol buffers 1151
group (SNMP) 1109
group change configuration command modes 62
group-specific queries (IGMP) 995

H

hardware dependent forwarding (multicast) 994
heartbeat interval (MLAG) 431
hello interval
OSPFv2 772
OSPFv3 845
hello message (PIM-SM) 1006
hello packet
OSPFv2 765
OSPFv3 839
hello-time (STP) 577
hello-time bridge timer (STP) 567
helper address (DHCP relay) 688
hierarchy, command modes 61
history buffer 55
history substitution 55
hold time
BGP 892
PIM-SM 1005
host (SNMP) 1110
host name
assigning 151
default 151
host routes
IPv4 686
IPv6 735
HTTP 45

I

IBGP..... *see* internal BGP
IEEE 802.1ad 31, 386
IEEE 802.1Q 31, 385
IGMP ..... *see* Internet Group Management Protocol
IGMP snooping
commands 1042–1074
configuration 1000–1003
description 995
filters 1000
IGMP profiles 1000
image file..... *see* EOS image
In Service Software Update, ISSU 30, 427
insufficient fan shutdown condition 234
interface configuration command modes 60
interface cost
OSPFv2 773
OSPFv3 846
interface status
OSPFv2 776
OSPFv3 848
interface-Ethernet configuration mode 254
interface-loopback configuration mode 336
interface-management configuration mode 254
interface-port channel configuration mode 290
interface-VLAN configuration mode 392
internal BGP, IBGP 889
internal neighbors (BGP) 891
internal ports 252
internal router, IR
OSPFv2 764
OSPFv3 838
internal spanning tree instance, ISTI 563
internal VLANs 387, 392
Internet Group Management Protocol, IGMP
commands 1024–1041
configuration 999–1000
description 995
enabling 998
versions 995
IGMP profiles .....*see* IGMP Snooping
snooping .....*see* IGMP Snooping
Internet Protocol, version 4, IPv4
address format 681
commands 689–726
configuration 683–688
default route 684
description 681–682
enabling 683
static routes 683
Internet Protocol, version 6, IPv6
address format 728
commands 736–761
configuration 730–735

default route .730
description .727–729
enabling unicast .730
static routes .730
interval, router advertisements (IPv6) 732
intra-area distance
OSPFv2 .768
OSPFv3 .842
IP access control list *see* access control list, ACL
IP address-mask 55
IP address-wildcard 56
IP route status
OSPFv2 .775
OSPFv3 .847
IPv4 *see* Internet Protocol, version 4
IPv6 *see* Internet Protocol, version 6
IPv4 access control list *see* access control list, ACL
IPv4 prefix list *see* prefix list
IPv6 access control list *see* access control list, ACL
IPv6 prefix list *see* prefix list
IPv6-ACL configuration command mode 460
isolated VLANs 387
ISSU *see* In Service Software Update
ISTI *see* internal spanning tree instance

J

join message (PIM-SM) 1007
jumbo frames 30

K

keepalive message/period
BGP .892
MLAG .425
keystrokes, cursor movement 54

L

LACP *see* Link Aggregation Control Protocol
LAG *see* Link Aggregation Group
LANZ *see* Latency Analyzer
LANZ protocol buffer schema 1151
last member query (IGMP) 1000
last member query response interval (IGMP) 995
Latency Analyzer, LANZ
commands .1153–1163
configuration .1146–1149
congestion events .1148
description .1145–1146
Google protocol buffers .1151
LANZ protocol buffer schema .1151
platform support .1146
streaming LANZ .1149–1152
layer 2 features 30
layer 3 features 31
layer 3 switching 325
learning mode (dynamic MAC address) 332
learning state (STP) 566
LEDs 234
Link Aggregation Control Protocol, LACP
commands .297–299, 309–316
configuration .290–292
description 30, 288
fallback 288, 291
modes 288
Link Aggregation Group, LAG
description 287
see also port channel
link debounce .269
Link Layer Discovery Protocol, LLDP 30
link state advertisements, LSA
description (OSPFv2) 764
description (OSPFv3) 838
LSA filtering (OSPFv2) 771
LSA overload (OSPFv2) 768
link state database, LSDB
OSPFv2 764
OSPFv3 838
link trap generation (SNMP) .1110
link-local address (IPv6) .728
Linux Bash CLI .29
Linux syntax .55
listening state (STP) .566
LLDP *see* Link Layer Discovery Protocol
load sharing .330, 684
local file (security) .96
local interface (MLAG) .430
local route preference (BGP) .893
local time .153
location string (SNMP) .1112
logging (ACL) .459, 464
login banner .155
loop guard (STP) .576
loopback interface .329, 336
LSA *see* link state advertisements

M

MAC access control list *see* access control list, ACL
MAC address
aging time 332
dynamic 332
static 328
MAC address table .328
displaying 333
MAC addresses
assignment 254
dynamic 328
static 331
MAC-ACL configuration command mode .460
main VRF .330
Management Information Base, MIB .1105
management interface
configuration 254
description 253
management port .29, 38, 53
manager (SNMP) .1105
mask, address .55
master router (VRRP) .531
match statements (route maps) .470
max-age (STP) .577
max-age bridge timer (STP) .567
max-hop (STP) .577
max-hop bridge timer (STP) .567
maximum transmission unit, MTU .329
mc-tx-queue configuration command mode .655

membership query (IGMP) 999
membership query interval (IGMP snooping) 1002
membership query response interval (IGMP snooping) . 1002
membership report (IGMP) 995
Message-Digest authentication (OSPFv2) 771
MET ..... *see* multicast expansion table
MIB ..... *see* Management Information Base
mirroring, port 327
MLAG ..... *see* Multi-Chassis Link Aggregation
MLAG ISSU 427
mode (STP) 568
model (switch) to platform map 59
modular ports, referencing 255
monitoring, port 327
more boot-config (command) 198
motd banner 155
MRIB ..... *see* multicast routing information base
mrouter ..... *see* multicast router
MSTI ..... *see* multiple spanning tree instance
MSTP ..... *see* Multiple Spanning Tree Protocol
MTU ..... *see* maximum transmission unit
MTU option suppression, router advertisement (IPv6) .. 733
multicast
  architecture 993–994
  control plane 993
  description 31
  forwarding plane 994
  routing 993, 998
multicast expansion table, MET 994
multicast router, mrouter (snooping IGMP) 995, 1003
multicast routing information base, MRIB 994
Multi-Chassis Link Aggregation, MLAG
  commands 442–454
  configuration 429–433
  description 30, 423–424
  examples 434–441
  MLAG configuration command mode 430
  restartability 426
multi-mode fiber (MMF) 249
multiple spanning tree instance, MSTI 563
Multiple Spanning Tree Protocol, MSTP 30, 562
multiplexing sessions (TACACS+) 99
multiprotocol BGP 890

N

native VLAN 390
neighbor activation (BGP) 898
Neighbor Discovery protocol (IPv6) 729, 731
neighbors
  OSPFv2 ..... 765, 778
  OSPFv3 ..... 839, 849
neighbors (BGP) 891
network ports (STP) 566
network route advertising (BGP) 899
Network Time Protocol, NTP
  configuring 153
  versions supported 180
network topology, fully meshed 893
next hop (BGP) 934, 935
normal area
  OSPFv2 ..... 765, 769
  OSPFv3 839
normal area (OSPFv3) 843
normal ports (STP) 566
notifications (SNMP) 1107
not-so-stubby-area, NSSA area (OSPFv2) 765, 769
NSSA area ..... *see* not-so-stubby-area
NTFS file system 197
NTP ..... *see* Network Time Protocol
null0 interface 329

O

Open Shortest Path First, version 2, OSPFv2
  commands 790–835
  configuration 767–780
  database 776
  description 31, 763–766
  displaying status 775
  examples 781–789
  neighbors 778
  redistributing routes 818
  router-OSPF configuration command mode 767
Open Shortest Path First, version 3, OSPFv3
  commands 860–887
  configuration 841–850
  database 849
  description 31, 837–840
  displaying status 847
  examples 851–859
  neighbors 849
  redistributing routes 879
  router-OSPFv3 configuration command mode ... 841
optical fiber classifications 249
OSPFv2 ..... *see* Open Shortest Path First, version 2
OSPFv3 ..... *see* Open Shortest Path First, version 3
out delay (BGP) 936
overheating shutdown condition 233
override hardware condition
  automatic fan speed 236
  insufficient fan shutdown 235
  overheating shutdown 235

P

parent modes 61
passive interface
  OSPFv2 769
  OSPFv3 842
password
  clear text 96
  enable 97
  encryption 96
  root account 98
  username 96
path cost (STP) 573
peer address (MLAG) 431
peer link (MLAG) 424, 431
peer switches (MLAG) 424
Per-VLAN Rapid Spanning Tree (PVRST+) 30
Petra platform switch models 59
PHY
  description 251
  displaying 261
physical interfaces 254
physical layer 251

PIM-SM . . . *see* Protocol Independent Multicast-Sparse Mode
plain text . . . . . *see* clear text
platform 59
platform to switch model map 59
point-to-point ports (STP) 566
port
  console (serial) 36
  DCS-7050Q-16 configuration 257
  Ethernet management 29, 38, 53
  RADIUS 102
  TACACS+ 99
  USB 67
port activity states (STP) 566
port channel interface
  commands 296, 302–308, 317–322, 396
  configuration 290
  description 287
port channel, description 287
port groups 257
port mirroring 29, 327, 334
port priority (STP) 569, 574
port security 30, 329
port settings (console, serial) 36
port shaping 639
port trust (QoS) 646
port type, displaying 260
portfast (STP) 574
power supplies
  description 234
  redundancy 234
  viewing status 238
preemption (VRRP) 531
prefix inclusion, router advertisement (IPv6) 732
prefix list
  commands 488, 496, 500, 509, 517, 518, 524, 526
  configuration 476–478
  description 476
  IPv6-pfx configuration command mode 477
  using 478
prefix, address 55
primary IP address (VRRP) 535
primary VLANs 391
priority (STP) 569
priority (VRRP) 531
Priority Flow Control, PFC 30
private VLANs 391
privilege level, authorization 114
Privileged EXEC command mode 60
prompts
  command modes 183
  description and configuration 156
Protocol Independent Multicast-Sparse Mode, PIM-SM
  commands 1075–1103
  configuration 1004–1007
  description 996–997
  enabling 998
protocol specific command modes 60
provisioning the switch
  manual 36
  automatic . . . . . *see* Zero Touch Provisioning
proxy ARP 682
prune message (PIM-SM) 1007

Q

q-in-q network 386
QoS . . . . . *see* Quality of Service
QSFP+ modules
  description 251–252
  Ethernet interface configuration 255
  ethernet interface configuration 257
  switch models 255
Quality of Service, QoS
  commands 652–680
  configuration 646–651
  CoS rewrite 638
  data fields 637
  description 30, 637–645
  platform specific implementations 640–645
  port settings 637
  port shaping 638
  port trust 646
  traffic classes 638
  transmit queues 638
querier (IGMP snooping) 1001
querier address (IGMP snooping) 1002
queriers (IGMP) 995
question mark . . . . . *see* ?
queue shaping 639
queue-monitor streaming configuration mode 1149

R

RADIUS . . . . . *see* Remote Authentication Dial In User Service
Rapid Per-VLAN Spanning Tree Protocol, Rapid-PVST . . 562
Rapid Spanning Tree Protocol, RSTP 30, 562
Rapid-PVST . . . *see* Rapid Per-VLAN Spanning Tree Protocol
rate limit, BPDU (STP) 579
reachable time, router advertisement (IPv6) 731
recovery procedures 41–44, 203
redistributing routes
  BGP 953
  OSPFv2 818
  OSPFv3 879
  RIP 984
redistributing static routes
  OSPFv2 769
  OSPFv3 843
redundancy, power supplies 234
region (MSTP) 562
regular expressions 56
reload (command) 46
reload delay period (MLAG) 432
Remote Authentication Dial In User Service, RADIUS 30, 100
rendezvous point, RP (PIM-SM) 996, 997, 1004
restartability 426
retransmit (RADIUS) 101
retransmit interval
  OSPFv2 772
  OSPFv3 845
reverse path forwarding, RPF (PIM) 993
RIP . . . . . *see* Routing Information Protocol
robustness variable (snooping IGMP) 1003
root account 98
root bridge, RB (STP) 564
root guard (STP) 576
root port, RP (STP) 565
round robin queue (Qos) 639

route aggregation (BGP) 896
route assignments (OSPFv2) 770
route distinguishers, RD 330
route map
clauses 470
commands 483, 503, 516–517, 519, 522, 528
configuration 472–474
creating 472
description 470–471
route-map configuration command mode 472
using 475
using (BGP) 894
route preference, local (BGP) 893
route redistribution 330
route reflection (BGP) 893
route summaries (OSPF) 770
route tag 683
route targets (BGP) 896
routed port 328, 334
router advertisment (IPv6) 731
router dead interval
OSPFv2 765
OSPFv3 839
router ID
BGP 954
OSPFv2 767
OSPFv3 841
router lifetime, router advertisement (IPv6) 732
router preference (IPv6) 734
router priority
OSPFv2 773
OSPFv3 846
router-BGP configuration command mode 891
router-OSPF configuration command mode 767
router-OSPF3 configuration command mode 841
router-RIP configuration mode 976
routing 325
Routing Information Protocol, RIP
commands 979–989
configuration 976–978
description 31, 975
redistributing routes 984
router-RIP configuration command mode 976
timers 977
routing table
IPv4 684
IPv6 734
routing tables 330
RP . . . . . *see* rendezvous point
RP tree (PIM-SM) 996
RSTP . . . . . *see* Rapid Spanning Tree Protocol
rule tracking (ACL) 459
running-config
description 201
displaying 63
saving 63, 64

S

sample rate (sFlow) 1187
scheduler, CLI 57
SCP 45
secondary addresses (VRRP) 534
secondary VLANs 391
secure shell, SSH
accessing EOS 53
connection management 40
description 29
serial port 36, 53
server access keys 96
server group (AAA)
description 103
server-group-radius configuration mode 118
server-group-tacacs+ configuration mode 119
service list (AAA) 104
service provider VLAN (q-in-q network) 386
session (VM Tracer) 1166
set statements (route maps) 470
sFlow
commands 1188–1199
configuration 1186–1187
description 1183–1185
SFP+ modules
description 251–252
ethernet interface configuration 257
shaping
ports 639
queue 639
shared ports (STP) 566
shortest path tree (SPT) 997
show boot-config (command) 198
show clock (command) 154
show history (command) 55
show ip route (command) 779, 850
show ntp associations (command) 154
show ntp status (command) 154
show reload cause (command) 203
show startup-config (command) 64
show tacacs (command) 100
show version (command) 46
shutdown condition
insufficient fans 234
overheating 233
Simple Network Management Protocol, SNMP
commands 1116–1144
configuration 1108–1115
description 29, 1105–1108
SNMP agent 1105
SNMP agent extending 1112
SNMP manager 1105
simple password authentication 771
single-mode fiber (SMF) 249
site of origin (BGP) 896
SNMP . . . . . *see* Simple Network Management Protocol
snooping querier (IGMP snooping) 1001
snooping, IGMP . . . . . *see* IGMP snooping
software image . . . . . *see* EOS image
solicited-node multicast address (IPv6) 729
source specific multicast (PIM) 1064, 1093
Spanning Tree Protocols, STP
description 30, 561–580
disabling 564
speed 258
SSH . . . . . *see* secure shell, SSH
SSH-management command mode 40
standard access control list . . . . . *see* access control list, ACL
standard communities (BGP) 895
startup query (IGMP) 999
startup-config

commands 201
definition 201
deleting 201
reverting 42
saving running-config 64
ZTP 35
state machine (BGP) 889
static groups (IGMP) 1000
static MAC address 328, 331
static route redistribution
OSPFv2 769
OSPFv3 843
static routes
IPv4 683
IPv6 730
static routing 31, 325
static RPs (PIM-SM) 997, 1004
std-ACL configuration command mode 460
std-IPv6-ACL configuration command mode 460
store and forward (control plane) 375
store and forward (data plane) 326, 338
storm control
commands 373
configuration 335
description 30, 327
STP ..... *see* Spanning Tree Protocols
STP agent restartablility 426
streaming LANZ ..... *see* Latency Analyzer
strict priority queue (Qos) 639
stub area
OSPFv2 .....765, 769
OSPFv3 .....839, 843
summary route default cost
OSPFv2 771
OSPFv3 844
suppression, router advertisement (IPv6) 733
SVI ..... *see* VLAN interface
SWI ..... *see* EOS image
switch file management 29
switch model to feature availability table 32
switch platform 59
switched port 328, 334
switched virtual interface ..... *see* VLAN interface
switching 324
switching, layer 3 325
switchport interface pairs 563
syntax assistance 55
system clock 153
system status, viewing 238

T

TACACS+
see Terminal Access Controller Access-Control System Plus
tag, route 683
tcpdump 29
Telnet 40, 53
Telnet-management command mode 40
temperature controls 233
temperature status, viewing 237
Terminal Access Controller Access-Control System Plus, TACACS+ 30, 98
timeout
RADIUS 101
TACACS+ 99
timers (RIP) 977
traceroute 29
traffic classes (QoS) 638
transceivers 251
transmission delay
OSPFv2 773
OSPFv3 846
transmit hold-count (STP) 578
transmit queues (QoS) 638
Trident platform siwtch models 59
truncated commands 54
trunk groups 386
trunk list (VLANs) 390
trunking 386
tx-queue configuration command mode 648, 679

U

uc-tx-queue configuration command mode 680
upgrades, EOS image 45
USB flash drive
configuration restoration 44
contents 67
image transfer 45
user (SNMP) 1110
user defined VRF 330
username
admin 36
definition 96
passwords 96
unprotected 97

V

VARP ..... *see* Virtual Address Resolution Protocol
versions (SNMP) 1107
VFAT file system 197
Virtual Address Resolution Protocol, VARP
commands 543–545
configuration 535–536
description 532
example 541
virtual IP address (VARP) 536
Virtual Local Area Networks, VLAN
commands 394–421
community VLANs 387
configuration 388–392
description 31, 385–387
internal VLANs 387, 392
isolated VLANs 387
primary VLANs 391
private VLANs 391
secondary VLANs 391
trunking 386
virtual mac address (VARP) 536
virtual router group 531
virtual router identifier, VRID 531
Virtual Router Redundancy Protocol, VRRP 30
commands 546–560
configuration 533–535
description 30, 531
examples 537–540

VLAN. . . . . .*see* Virtual Local Area Networks
VLAN interface 387, 392
VM Tracer
commands 1170–1181
configuration 1166–1169
description 1165–1166
VM tracer mode 1167
vmtracer configuration command mode 1166
VM tracer mode 1167
vmtracer configuration mode 1166
VPN Routing and Forwarding, VRF 330
VRF
context aware commands 339
context-active 339
VRF . . . . . .*see* VPN Routing and Forwarding
VRRP . . . . . *see* Virtual Router Redundancy Protocol

W

wildcard, IP address 56
write memory (command) 63, 64

Z

Zero Touch Provisioning, ZTP
cancelling 37
definition 35
provisioning the switch 35
set up 204
ZTP. . . . . .*see* Zero Touch Provisioning