

Summary

Several mechanisms exist to manage and instrument Arista Networks' devices, ranging from industry standard SNMP counters to more proprietary platform-centric functionality and deep debugging capabilities. With specific regard to the 7100 family the following management tools are available:

1. Syslog and Console Logging
2. SNMP Versions 1,2 and 3
3. Hardware Specific 'show' Commands
4. System and Process Level Logging
5. TCPdump
6. EOS Process Tracing
7. Other Monitoring Tools (LANZ, sFlow, Port Mirroring)

This document serves to highlight the basic parameters required to automate monitoring of an EOS based device, while providing a high level overview of additional, more advanced functionality for low level troubleshooting and application specific monitoring.

Configuring Syslog and Console Logging

For common system logging, EOS follows industry standard configuration semantics:

```
7124SX(config)#logging ?
buffered Set buffered logging parameters
console Set console logging parameters
event Global events
facility Set logging facility
format Set logging format parameters
host Set syslog server IP address and parameters
level Configure logging severity
on Enable logging to all supported destinations
trap Set syslog server logging level
```

```
7124SX(config)#logging host logs.foo.com
```

Console logging can be enabled, if no specific severity level is specified console logging will default to error conditions only:

```
7124SX(config)#logging console ?
alerts Immediate action needed (severity=1)
critical Critical conditions (severity=2)
debugging Debugging messages (severity=7)
emergencies System is unusable (severity=0)
errors Error conditions (severity=3)
informational Informational messages (severity=6)
notifications Normal but significant conditions (severity=5)
warnings Warning conditions (severity=4)
<0-7> Logging severity level
<cr>
```

SNMP Configuration and Overview

EOS supports a growing number of both Arista proprietary and standards based MIBs providing the ability to quickly integrate devices into 3rd party monitoring solutions. The current list of supported MIBs can be found by visiting:

<http://www.aristanetworks.com/en/support/aristasnmpmibs>

Configuring SNMP support on the device follows industry standard (e.g. for SNMPv2)

```
7124SX>en
7124SX#conf t
7124SX(config)#snmp-server community public
7124SX(config)#snmp-server host trap.foo.com public
```

EOS also natively provides the ability to walk and search local MIBs enabling easy location of specific OIDs

```
7124SX(config)#sh snmp mib ?
get          get one object
get-next     get the next object
table       get the contents of a table
walk        walk a subtree

7124SX(config)#sh snmp mib walk ?
OID          An object-ID (e.g., IP-MIB::ipAddrTable)
>           Redirect output to URL
>>         Append redirected output to URL
|           Output modifiers
<cr>

7124SX(config)#sh snmp mib walk . | grep -i processor
HOST-RESOURCES-MIB::hrDeviceType[1] = OID: HOST-RESOURCES-TYPES::hrDeviceProcessor
HOST-RESOURCES-MIB::hrDeviceType[2] = OID: HOST-RESOURCES-TYPES::hrDeviceProcessor
HOST-RESOURCES-MIB::hrDeviceType[3] = OID: HOST-RESOURCES-TYPES::hrDeviceProcessor
HOST-RESOURCES-MIB::hrDeviceDescr[1] = STRING: AMD Turion(tm) II Neo N41L Dual-Core
Processor
HOST-RESOURCES-MIB::hrProcessorFrwID[1] = OID: SNMPv2-SMI::zeroDotZero
HOST-RESOURCES-MIB::hrProcessorFrwID[2] = OID: SNMPv2-SMI::zeroDotZero
HOST-RESOURCES-MIB::hrProcessorFrwID[3] = OID: SNMPv2-SMI::zeroDotZero
HOST-RESOURCES-MIB::hrProcessorLoad[1] = INTEGER: 6
HOST-RESOURCES-MIB::hrProcessorLoad[2] = INTEGER: 4
HOST-RESOURCES-MIB::hrProcessorLoad[3] = INTEGER: 7
7124SX(config)#
```

Suggested SNMP OIDs for General System Health

When monitoring the overall health of an EOS based device, both internal factors such as CPU and memory load and environmental factors are useful metrics. These figures are available both from the CLI and via SNMP with examples provided below.

CPU and Memory Monitoring

The 7100 series utilize dual-core CPUs, the status of which can be viewed quickly from the CLI:

```
7124SX(config)#show proc top
```

```
top - 09:55:32 up 33 days, 21:42, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 132 total, 1 running, 131 sleeping, 0 stopped, 0 zombie
Cpu(s): 3.5%us, 0.2%sy, 0.0%ni, 96.4%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 4039008k total, 1275948k used, 2763060k free, 104728k buffers
Swap: 0k total, 0k used, 0k free, 756980k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1393	root	20	0	277m	71m	32m	S	6.0	1.8	3031:12	FocalPoint
1368	root	20	0	181m	75m	40m	S	1.0	1.9	462:13.00	Sysdb
1366	root	20	0	161m	26m	2132	S	0.3	0.7	145:15.15	ProcMgr-worker
1369	root	20	0	182m	65m	34m	S	0.3	1.6	61:16.56	Fru
1384	root	20	0	163m	39m	13m	S	0.3	1.0	60:53.05	SuperServer
1394	root	20	0	175m	44m	18m	S	0.3	1.1	50:47.20	Lag+LacpAgent
1402	root	20	0	164m	37m	12m	S	0.3	1.0	213:48.07	Smbus
1405	root	20	0	164m	37m	12m	S	0.3	1.0	24:15.77	Mdio
1408	root	20	0	164m	38m	13m	S	0.3	1.0	19:10.05	PowerSupplyDete
1414	root	20	0	163m	35m	10m	S	0.3	0.9	92:50.65	Thermostat
12524	admin	20	0	20144	10m	8536	R	0.3	0.3	0:00.14	top

Under the HOST-RESOURCES MIB, the dual-core CPU appears as three distinct processors, the first providing an average view of the behavior of the two physical cores that follow. The values are percentages expressed as integers;

1st generation platforms (7124S, 7120T, 7148S, 7140T and 7148SX)

```
HOST-RESOURCES-MIB::hrDeviceDescr[1] = STRING: AMD Athlon(tm) 64 X2 Dual Core Processor 3400+
HOST-RESOURCES-MIB::hrDeviceDescr[2] = STRING: Core 1
HOST-RESOURCES-MIB::hrDeviceDescr[3] = STRING: Core 2
HOST-RESOURCES-MIB::hrProcessorLoad[1] = INTEGER: 22
HOST-RESOURCES-MIB::hrProcessorLoad[2] = INTEGER: 21
HOST-RESOURCES-MIB::hrProcessorLoad[3] = INTEGER: 23
```

2nd generation platforms (7124SX)

HOST-RESOURCES-MIB::hrDeviceDescr[1] = STRING: AMD Turion(tm) II Neo N41L Dual-Core Processor
HOST-RESOURCES-MIB::hrDeviceDescr[2] = STRING: Core 1
HOST-RESOURCES-MIB::hrDeviceDescr[3] = STRING: Core 2
HOST-RESOURCES-MIB::hrProcessorLoad[1] = INTEGER: 6
HOST-RESOURCES-MIB::hrProcessorLoad[2] = INTEGER: 4
HOST-RESOURCES-MIB::hrProcessorLoad[3] = INTEGER: 7

Memory utilization can be monitored using the following OIDs that provide the description, total amount of memory and its utilization, these items are common to all 7100 series devices.

HOST-RESOURCES-MIB::hrStorageDescr[1] = STRING: RAM
HOST-RESOURCES-MIB::hrStorageSize[1] = INTEGER: 4039008
HOST-RESOURCES-MIB::hrStorageUsed[1] = INTEGER: 1274724

Environmental Factors

Each device is equipped with an array of sensors covering temperature monitoring, fan speed and power availability. The detailed information available through the CLI maps directly to a number of OIDs.

1st generation platforms (7124S, 7120T, 7148S, 7140T and 7148SX)

```
Arista7124S#sh environment all
System temperature status is: Ok
```

Sensor	Description	Temperature	Alert Threshold	Critical Threshold
1	Front-panel temp sensor	34.750C	65C	75C
2	Fan controller 1 sensor	36.000C	75C	85C
3	Fan controller 2 sensor	40.000C	75C	85C
4	Switch chip 1 sensor	45.000C	105C	115C
5	VRM 1 temp sensor	64.000C	105C	110C

```
System cooling status is: Ok
```

```
Ambient temperature: 34C
```

```
Airflow: front-to-back
```

Fan Tray	Status	Speed
1	Ok	49%
2	Ok	49%
3	Ok	49%
4	Ok	49%
5	Ok	49%

Power Supply	Model	Capacity	Input Current	Output Current	Output Power	Status
1	PWR-650AC	650W	0.69A	11.00A	132.0W	Ok

```
Arista7124S#
```

The following ENTITY-MIB OIDs provide temperature monitoring relating to the sensors as listed. Each integer output is degrees Celsius x 10:

```
ENTITY-MIB::entPhysicalDescr[100006001] = STRING: Front-panel temp sensor
ENTITY-MIB::entPhysicalDescr[100006002] = STRING: Fan controller 1 sensor
ENTITY-MIB::entPhysicalDescr[100006003] = STRING: Fan controller 2 sensor
ENTITY-MIB::entPhysicalDescr[100006004] = STRING: Switch chip 1 sensor
ENTITY-MIB::entPhysicalDescr[100006005] = STRING: VRM 1 temp sensor
ENTITY-SENSOR-MIB::entPhySensorValue[100006001] = INTEGER: 348
ENTITY-SENSOR-MIB::entPhySensorValue[100006002] = INTEGER: 360
ENTITY-SENSOR-MIB::entPhySensorValue[100006003] = INTEGER: 400
ENTITY-SENSOR-MIB::entPhySensorValue[100006004] = INTEGER: 450
ENTITY-SENSOR-MIB::entPhySensorValue[100006005] = INTEGER: 640
```

Fan-speed is measured in RPM that is reflected in the CLI as a percentage of the maximum nominal speed of 27000rpm:

```
ENTITY-MIB::entPhysicalDescr[100601110] = STRING: Fan Tray 1 Fan 1  
ENTITY-MIB::entPhysicalDescr[100602110] = STRING: Fan Tray 2 Fan 1  
ENTITY-MIB::entPhysicalDescr[100603110] = STRING: Fan Tray 3 Fan 1  
ENTITY-MIB::entPhysicalDescr[100604110] = STRING: Fan Tray 4 Fan 1  
ENTITY-MIB::entPhysicalDescr[100605110] = STRING: Fan Tray 5 Fan 1
```

```
ENTITY-SENSOR-MIB::entPhySensorValue[100601111] = INTEGER: 12690  
ENTITY-SENSOR-MIB::entPhySensorValue[100602111] = INTEGER: 12690  
ENTITY-SENSOR-MIB::entPhySensorValue[100603111] = INTEGER: 12420  
ENTITY-SENSOR-MIB::entPhySensorValue[100604111] = INTEGER: 12690  
ENTITY-SENSOR-MIB::entPhySensorValue[100605111] = INTEGER: 12420
```

2nd generation platforms (7124SX):

```
7124SX(config)#sh environment all
System temperature status is: 0k
```

Sensor	Description	Temperature	Alert Threshold	Critical Threshold
1	Cpu temp sensor	38.106C	95C	100C
2	Rear temp sensor	30.750C	55C	65C
3	Front temp sensor	30.000C	55C	65C
4	Board temp sensor	29.625C	90C	110C
5	Bali temp sensor	44.875C	85C	105C

```
System cooling status is: 0k
Ambient temperature: 30C
Airflow: front-to-back
```

Fan Tray	Status	Speed
1	0k	75%
2	0k	75%
3	0k	75%
4	0k	75%
PowerSupply1	0k	80%
PowerSupply2	0k	80%

Power Supply	Model	Capacity	Input Current	Output Current	Output Power	Status
1	PWR-460AC-F	460W	0.00A	0.00A	0.0W	AC Loss
2	PWR-460AC-F	460W	0.41A	7.50A	91.0W	0k

```
7124SX(config)#
```

The following ENTITY-MIB OIDs provide temperature monitoring relating to the sensors as listed. Each integer output is degrees Celsius x 10:

```
ENTITY-MIB::entPhysicalDescr[100006001] = STRING: Cpu temp sensor
ENTITY-MIB::entPhysicalDescr[100006002] = STRING: Rear temp sensor
ENTITY-MIB::entPhysicalDescr[100006003] = STRING: Front temp sensor
ENTITY-MIB::entPhysicalDescr[100006004] = STRING: Board temp sensor
ENTITY-MIB::entPhysicalDescr[100006005] = STRING: Bali temp sensor
ENTITY-SENSOR-MIB::entPhySensorValue[100006001] = INTEGER: 381
ENTITY-SENSOR-MIB::entPhySensorValue[100006002] = INTEGER: 308
ENTITY-SENSOR-MIB::entPhySensorValue[100006003] = INTEGER: 300
ENTITY-SENSOR-MIB::entPhySensorValue[100006004] = INTEGER: 296
ENTITY-SENSOR-MIB::entPhySensorValue[100006005] = INTEGER: 449
```

Fan-speed is measured in RPM that is reflected in the CLI as a percentage of the maximum nominal speed of 17820rpm:

```
ENTITY-MIB::entPhysicalDescr[100601110] = STRING: Fan Tray 1 Fan 1
ENTITY-MIB::entPhysicalDescr[100602110] = STRING: Fan Tray 2 Fan 1
ENTITY-MIB::entPhysicalDescr[100603110] = STRING: Fan Tray 3 Fan 1
ENTITY-MIB::entPhysicalDescr[100604110] = STRING: Fan Tray 4 Fan 1
```

```
ENTITY-SENSOR-MIB::entPhySensorValue[100601111] = INTEGER: 13320
ENTITY-SENSOR-MIB::entPhySensorValue[100602111] = INTEGER: 13500
ENTITY-SENSOR-MIB::entPhySensorValue[100603111] = INTEGER: 13320
ENTITY-SENSOR-MIB::entPhySensorValue[100604111] = INTEGER: 13860
```

Suggested SNMP OIDs for Interface Statistics

A selection of standard MIBs cater for a multitude of interface stats counters covering throughput, packet size and error statistics. Using the integrated MIB browsing capability it is possible to select appropriate counters from MIBs such as:

EtherLike-MIB
IF-MIB
RMON-MIB

For example:

```
Arista7124SX#sh snmp mib walk . | grep IF-MIB::ifHC
IF-MIB::ifHCInOctets[1] = Counter64: 0
IF-MIB::ifHCInOctets[2] = Counter64: 0
IF-MIB::ifHCInOctets[3] = Counter64: 397019758
IF-MIB::ifHCInOctets[4] = Counter64: 221921789
IF-MIB::ifHCInOctets[5] = Counter64: 13137588
:
```

Platform Specific Show Commands

Under the EOS CLI, a hierarchy of hardware specific show commands enable granular visibility into detailed hardware counters. The 7100 series utilizes the Fulcrum FM4000 ‘Bali’ ASIC, which provides the naming convention in the command tree.

Reviewing the context help for ‘sh platform fm4000’ below there are a few areas of immediate interest for scripted monitoring and troubleshooting as well as access to the hardware MAC table, (m)route cache and mirroring groups.

```
7124SX#sh platform fm4000 ?
  bali1          bali1 switch
  interface      Show interface-specific info
  mac-address-table Show hardware MAC address table
  mirror-groups  Show internal mirror session info
  <cr>
```

NB: Depending on the specific platform, the ASIC(s) may be referred to as ‘fp’ (7124S and 7120T), bali1 (7124SX), bali1..3 (7148S and 7140T) and in the 7148SX line0..3 and fabric0..1 for port and fabric ASICs respectively.

‘show platform fm4000 int e1 counters verbose’ for example, provides a large number of metrics including a packet size histogram, drop counters and errors:

Outputs for all interfaces can be collected using ‘show platform fm4000 bali1 counters verbose’ while the ‘| nz’ filter may be used to remove lines with zero value counters.

```
7124SX#sh platform fm4000 int e 1 counters verbose
* These counters are either 8, 16 or 32 bits and can wrap around
# These counters reset to zero when port changes speed
Switch bali1 port 2 (interface Ethernet1):
rxUcstPktsNonIP: 0
rxUcstPktsIPv4: 0
rxUcstPktsIPv6: 0
rxBcstPktsNonIP: 0
rxBcstPktsIPv4: 0
rxBcstPktsIPv6: 0
:
```

Understanding TCAM utilization may also be critical in planning for topology and configuration changes. The Fulcrum ASIC’s TCAM is shared between (unicast and multicast) routing and ACL definitions and the balance between these and free space is available via the following command:

```
7124SX#sh platform fm4000 bali1 tcam usage
```

Slice Function	Util(%)	Used	Free
0-3 Reserved	9	24	232
VRRP	0	0	256
4 Unused	0	0	512
5 Unused	0	0	512
6 Unused	0	0	512
7 Unused	0	0	512
8 Unused	0	0	512
9 Unused	0	0	512
10 Unused	0	0	512
11 Unused	0	0	512

```
:
```

System and Process Logging

In addition to the EOS log provided by the 'show log' CLI command, EOS keeps detailed system-wide logs and individual agent process logs for multiple agent instances (due to reconfiguration or in-service stateful repair). These logs can be found in the underlying Linux shell as follows:

```
7124SX#bash sudo tail /var/log/messages
Sep  1 13:01:01 se302 run-parts(/etc/cron.hourly)[16209]: starting logrotate
Sep  1 13:01:01 se302 run-parts(/etc/cron.hourly)[16227]: finished logrotate
Sep  1 14:01:01 se302 CROND[17331]: (root) CMD (run-parts /etc/cron.hourly)
Sep  1 14:01:01 se302 run-parts(/etc/cron.hourly)[17331]: starting 0anacron
Sep  1 14:01:01 se302 anacron[17340]: Anacron started on 2011-09-01
Sep  1 14:01:01 se302 anacron[17340]: Normal exit (0 jobs run)
Sep  1 14:01:01 se302 run-parts(/etc/cron.hourly)[17342]: finished 0anacron
Sep  1 14:01:01 se302 run-parts(/etc/cron.hourly)[17331]: starting logrotate
Sep  1 14:01:01 se302 run-parts(/etc/cron.hourly)[17349]: finished logrotate
Sep  1 14:37:34 se302 Cli: %SYS-5-CONFIG_I: Configured from console by admin on vty3 (172.22.6.197)
7124SX#
```

bash shell commands may be executed directly from the CLI or alternatively a shell may be launched providing full access to familiar Linux tool sets for managing files:

```
7124SX#bash
```

```
Arista Networks EOS shell
```

```
[admin@7124SX ~]$ cd /var/log
[admin@7124SX log]$ sudo cat messages |grep -i stp
Jul 29 12:14:10 localhost Launcher: %LAUNCHER-6-PROCESS_START: Configuring process 'StpTopology' to
start in role 'ActiveSupervisor'
Jul 29 12:14:10 localhost Launcher: %LAUNCHER-6-PROCESS_START: Configuring process 'Stp' to start in
role 'ActiveSupervisor'
Jul 29 12:14:10 localhost ProcMgr-worker: %PROCMgr-6-PROCESS_STARTED: 'StpTopology' starting with
PID=1404 (PPID=1366) -- execing '/usr/bin/StpTopology'
Jul 29 12:14:10 localhost ProcMgr-worker: %PROCMgr-6-PROCESS_STARTED: 'Stp' starting with PID=1410
(PPID=1366) -- execing '/usr/bin/Stp'
:
```

Individual agent logs are available in '/var/log/agents' multiple restarts of an agent will create multiple files, each suffixed with the new process ID.

```
[admin@Arista7124SX ~]$ cd /var/log/agents
[admin@Arista7124SX agents]$ ls
Aaa-1391          Ira-1388          PciBus-1412      Rib-14549        SuperServer-1384
Acl-1411         Lag-1394          PhyAeluros-1438 Sb820-1395      Sysdb-1368
Arp-1413         Launcher-1367     PhyEthtool-1401 Scd-1416         Thermostat-1414
Ebra-1419        LedPolicy-1389   Pmbus-1662       Sflow-14045     TopoAgent-1417
FanDetector-1387 Lldp-1383        PowerManager-1403 Sflow-14082     Ucd9012-1385
FocalPoint-1393 Lm73-1418        PowerSupplyDetector-1408 Smbus-1402      Xcvr-1409
Fru-1369         Lm95234-1390    ProcMgr-worker-1366 Snmp-1386
FruSnmp-1407     Mdio-1405        Rib-1420         Stp-1410
IcmpSnooping-1421 MirroringAgent-1406 Rib-14481        StpTopology-1404
[admin@Arista7124SX agents]$
```

On occasion it may be necessary to collect the contents of the agent logs for TAC, the simplest way to group all the logs together onto the flash is:

```
7124SX#bash cat /var/log/agents/* >/mnt/flash/agents.log
7124SX#dir
Directory of flash:/

-rwx   220242219      Jul 7 05:44  EOS-4.7.3.swi
-rwx     4877        Sep 1 15:11  agents.log
-rwx     25         Jul 14 08:15  boot-config
drwx   4096         Jul 14 08:17  debug
drwx   4096         Sep 1 14:45  persist
-rwx   2053         Jul 29 15:07  startup-config

1852211200 bytes total (389935104 bytes free)
7124SX#
```

As with regular CLI commands, shell commands may be added to aliases for easy repetition:

```
7124SX(config)#alias getlogs bash cat /var/log/agents/* >/mnt/flash/agents.log

7124SX(config)#sh alias
getlogs bash cat /var/log/agents/* >/mnt/flash/agents.log

7124SX(config)#getlogs
7124SX(config)#dir flash:a*
Directory of flash:/a*

-rwx     4877        Sep 1 15:14  agents.log

1852211200 bytes total (389935104 bytes free)
7124SX(config)#
```

Using tcpdump to Monitor Control Plane Traffic

The Linux tcpdump utility is packaged with EOS allowing fast and efficient monitoring of control plane traffic. tcpdump provides ready access to L2/3 protocols and any other traffic destined for the switch itself without the need to SPAN interfaces.

To tcpdump an interface, first find out the Linux name for the interface (note, L2, L3 and Management interfaces are listed individually):

```
7124SX#bash ifconfig
cpu      Link encap:Ethernet  HWaddr 00:1C:73:0F:01:00
         UP BROADCAST RUNNING MULTICAST  MTU:9216  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

et1      Link encap:Ethernet  HWaddr 00:1C:73:0F:01:00
         UP BROADCAST MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

:
:
ma1      Link encap:Ethernet  HWaddr 00:1C:73:0F:00:FF
         inet addr:172.22.26.166  Bcast:255.255.255.255  Mask:255.255.254.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:4218484 errors:0 dropped:0 overruns:0 frame:0
         TX packets:13768 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:1574843314 (1.4 GiB)  TX bytes:3219623 (3.0 MiB)
         Interrupt:21

vlan10   Link encap:Ethernet  HWaddr 00:1C:73:0F:01:00
         UP BROADCAST MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Next, from either the CLI or the bash shell, run the utility passing the required interface and optionally a standard filter:

```
7124SX#bash tcpdump -i et1 arp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on et1, link-type EN10MB (Ethernet), capture size 65535 bytes
15:25:03.516271 00:1b:21:29:d7:f2 (oui Unknown) > Broadcast, ethertype ARP (0x0806), length 60:
Request who-has dc1-tac-rack4-serial1.aristanetworks.com tell lab.aristanetworks.com, length 46
15:25:04.516296 00:1b:21:29:d7:f2 (oui Unknown) > Broadcast, ethertype ARP (0x0806), length 60:
Request who-has dc1-tac-rack4-serial1.aristanetworks.com tell lab.aristanetworks.com, length 46
15:25:05.516309 00:1b:21:29:d7:f2 (oui Unknown) > Broadcast, ethertype ARP (0x0806), length 60:
Request who-has dc1-tac-rack4-serial1.aristanetworks.com tell lab.aristanetworks.com, length 46
15:25:05.551091 00:25:11:06:87:cb (oui Unknown) > Broadcast, ethertype ARP (0x0806), length 60:
Request who-has dc1-tac-rack4-serial1.aristanetworks.com tell lab.aristanetworks.com, length 46
^C
5 packets captured
5 packets received by filter
0 packets dropped by kernel
7124SX#
```

Tracing Processes with EOS

EOS provides operators with extensive troubleshooting tools to help debug control plane and protocol layer interactions through built-in tracing that delivers live debug output to the CLI. To configure tracing, first review the available agent processes:

```
7124SX#sh trace ?
Aaa                Aaa agent
Ac1                Ac1 agent
Adt7462Agent       Adt7462Agent agent
Arp                Arp agent
Cdp                Cdp agent
Dcbx               Dcbx agent
DhcpRelay          DhcpRelay agent
Ebra               Ebra agent
ElectionMgr        ElectionMgr agent
FPLanz             FPLanz agent
FanDetector        FanDetector agent
FileReplicator     FileReplicator agent
FocalPoint         FocalPoint agent
FrameBufferAgent  FrameBufferAgent agent
:
```

Having selected an agent to trace, review the available trace facilities for that process:

```
7124SX#sh trace Arp
Global trace settings for agent Arp
-----
Tracing sent to stderr

date:      enabled
time:      enabled
PID:       enabled
facility name: enabled (width 20)
trace level: enabled
filename:   disabled (width 20)
line number: disabled
function name: disabled (width 20)
object name: disabled (width 20)

Trace facility settings for agent Arp is
-----
Activity      enabled .....
Agent         enabled .....
AgentBase     enabled .....
AgentEnv      enabled .....
ArpRefresher  enabled .....
ArpRefresherInputSm enabled .....
ArpResolver   enabled .....
CEntityManager enabled .....
Clock         enabled .....
Dir           enabled .....
Entity        enabled .....
:
```

Finally, configure a temporary file to store the trace information (stored in /tmp) and enable tracing for each required facility (or * for all facilities) and level. Once complete run 'trace monitor' to output live process trace information to the CLI:

```
7124SX(config)#trace Arp ?
enable      Enable tracing
field-width Set the width of fields in the trace output
filename    Set the filename for trace output
setting     Enable tracing with raw setting
show       Enable fields to be included in the trace output

7124SX(config)#trace Arp filename arp.trace
7124SX(config)#trace Arp enable ?
WORD Trace facility name

7124SX(config)#trace Arp enable * ?
all      Enable tracing at all levels
levels  Enable tracing at one or more levels

7124SX(config)#trace Arp enable * all
7124SX(config)#trace monitor Arp
2011-09-01 15:36:32.141743 1413 Tac::ActivityImpl 9 File descriptor 5 has 2 references
2011-09-01 15:36:32.141770 1413 Tac::ActivityImpl 9 File descriptor 8 has 2 references
2011-09-01 15:36:32.141800 1413 Tac::ActivityImpl 9 File descriptor 11 has 2 references
2011-09-01 15:36:32.141827 1413 Tac::ActivityImpl 9 File descriptor 9 has 2 references
2011-09-01 15:36:32.141857 1413 Tac::ActivityImpl 9 File descriptor 14 has 2 references
2011-09-01 15:36:32.141887 1413 Tac::ActivityImpl 8 selectTime=inf executionTime=inf
stopOnQuiesce=0 requiredSelect=1
2011-09-01 15:36:32.141925 1413 Tac::ActivityImpl 7 Computing poll time 2949824.754416290205
2949794.755470665172
2011-09-01 15:36:32.141955 1413 Tac::ActivityImpl 7 Polling file descriptors for 29.998945625
2011-09-01 15:36:32.141982 1413 Tac::ActivityImpl 9 fastForward is set to 0
2011-09-01 15:36:32.142007 1413 Tac::ActivityImpl 6 Selecting

^C
7124SX(config)#
```

Other Monitoring Tools

The 7100 series support additional features for improving network visibility including sFlow (RFC3176) , LANZ and Port Mirroring.

LANZ

Arista's Latency ANalyZer (LANZ) provides the unique ability to monitor switch queue-depth on a per-port basis with microsecond granularity. LANZ can provide early warning of impending congestion and increasing latency through CLI, Syslog and also an application layer streaming export protocol providing administrators and applications themselves real-time awareness of changing network conditions and microburst behavior.

sFlow

sFlow is an embedded sampling technology designed to facilitate high rate traffic and statistics export from network devices with no impact to forwarding performance. sFlow samples may be sent to a collector application supporting a specific requirement (visualization, modeling, troubleshooting, capacity planning, IDS) or may also be converted to pcap data or NetFlow for consumption in other applications.

```
7124SX(config)#sflow ?
destination      Set the collector IP address
polling-interval Set polling interval (secs) for sFlow
run              Run sFlow globally
sample          Set sample rate for sFlow
source           Set the source IP address
source-interface Configure the source interface for sFlow datagrams
```

```
7124SX(config)#sflow destination 192.168.1.65
```

```
7124SX(config)#sflow run
```

```
7124SX(config)#sh sflow int
```

```
7124SX(config-if-Et1-24)#
```

```
7124SX(config-if-Et1-24)#sh sflow int
```

```
sFlow Interface (s):
```

```
-----
```

```
Ethernet1
```

```
Ethernet2
```

```
Ethernet3
```

```
:
```

Port Mirroring

Port Mirroring is used on a switch to send a copy of packets seen on one or more ports to a network monitoring connection on another switch port. This is commonly used for network appliances that require monitoring of network traffic like an intrusion-detection system.

```
7124SX(config)#monitor session MON1 destination e1
7124SX(config)#monitor session MON1 source e4,6-7,10,12-$
7124SX(config)#sh monitor session
```

```
Session MON1
-----
```

```
Source Ports
```

```
Both:      Et6, Et7, Et10, Et12, Et13, Et14
           Et15, Et16, Et17, Et18, Et19, Et20
           Et21, Et22, Et23, Et24
```

```
Destination Port: Et1
```

```
7124SX(config)#
```