OmniSwitch 6860/6900/10K Troubleshooting Guide

Alcatel-Lucent Enterprise
26801 West Agoura Road, Calabasas, CA 91301
(818) 880-3500
www.alcatel-lucent.com

Table of Contents

1. About This Guide

The *OmniSwitch troubleshooting* guide describes how to use Command Line Interface (CLI) and low level shell commands available on the OmniSwitch Family to troubleshoot switch and network problems. Reading the OmniSwitch User Guides prior to reading this guide is highly recommended.

This document is for internal Alcatel-Lucent employees only. Distribution to clients, users, and partners should ONLY be done with the consent of Technical Support.

1.1. **Supported Platforms**

This information in this guide applies to the following products:

- OmniSwitch 6860
- OmniSwitch 6860E
- OmniSwitch 6900
- OmniSwitch 10K

1.2. **Who Should Read this Manual?**

The principal audience for this user guide is Service and Support personnel who need to troubleshoot switch problems in a live network. In addition, network administrators and IT support personnel who need to configure and maintain switches and routers can use this guide to troubleshoot a problem upon advice from Alcatel-Lucent Service and Support personnel. However, this guide is *not* intended for novice or first-time users of Alcatel-Lucent OmniSwitches. Misuse or failure to follow procedures in this guide correctly can cause lengthy network down time and/or permanent damage to hardware. Caution must be followed on distribution of this document.

1.3. **When Should I Read this Manual?**

Always read the appropriate section or sections of this guide *before* you log into a switch to troubleshoot problems. Once you are familiar with the commands and procedures in the appropriate sections you can use this document as reference material when you troubleshoot a problem.

1.4. **What is in this Manual?**

The principal sections (i.e., the chapters numbered numerically) use CLI and Dshell commands to analyze and troubleshoot switch problems. Each section documents a specific switch feature (e.g., hardware, server load balancing, routing).

Note. Dshell commands should only be used by Alcatel-Lucent personnel or under the direction of Alcatel-Lucent. Misuse or failure to follow procedures that use Dshell commands in this guide correctly can cause lengthy network down time and/or permanent damage to hardware.

1.5. **What is Not in this Manual?**

This guide is intended for troubleshooting switches in live networks. It does not provide step-by-step instructions on how to set up particular features on the switch or a comprehensive reference to all CLI commands available in the OmniSwitch. For detailed syntax on non debug CLI commands and comprehensive information on how to configure particular software features in the switch, consult the user guides, which are listed in "Related Documentation" on page 4.

1.6. **How is the Information Organized?**

Each chapter in this guide includes troubleshooting guidelines related to a single software feature, such as server load balancing or link aggregation.

1.7. **Related Documentation**

The following are the titles and descriptions of all the Release 8 and later OmniSwitch user guides:
• *OmniSwitch 6860/6860E Hardware Users Guide*
Complete technical specifications and procedures for OmniSwitch 6860, power supplies, fans, and Network Interface (NI) modules.
• *OmniSwitch 6900 Hardware Users Guide*
Complete technical specifications and procedures for OmniSwitch 6900, power supplies, fans, and Network Interface (NI) modules.
• *OmniSwitch 10K Hardware Users Guide*
Complete technical specifications and procedures for OmniSwitch 10K, power supplies, fans, and Network Interface (NI) modules.
• *OmniSwitch AOS Release(7/8) CLI Reference Guide*
Complete reference to all CLI commands supported on the OmniSwitch. Includes syntax definitions, default values, examples, usage guidelines and CLI-to-MIB variable mappings.
• *OmniSwitch AOS Release(7/8) Switch Management Guide*
Includes procedures for readying an individual switch for integration into a network. Topics include the software directory architecture, image rollback protections, authenticated switch access, managing switch files, system configuration, using SNMP, and using web management software (WebView).
• *OmniSwitch AOS Release (7/8) Network Configuration Guide*
Includes network configuration procedures and descriptive information on all the major software features and protocols included in the base software package. Chapters cover Layer 2 information (Ethernet and VLAN configuration), Layer 3 information (RIP and static routes), security options (authenticated VLANs), Quality of Service (QoS), and link aggregation.
• *OmniSwitch AOS Release (7/8) Series Advanced Routing Configuration Guide*
Includes network configuration procedures and descriptive information on the software features and protocols included in the advanced routing software package (OSPF, IS-IS,BGP, DVMRP, PIM-SM).
OmniSwitch AOS Release 7 Data Center Switching Guide
Includes configuration information for data center networks using virtualization technologies (SPBM and UNP), Data Center Bridging protocols (PFC, ETC, and DCBX), and FCoE/FC gateway functionality.
• *OmniSwitch AOS Release (7/8) Tranceiver Guide*
This OmniSwitch Transceivers Guide provides specifications and compatibility information for the supported OmniSwitch transceivers for all OmniSwitch AOS 8 Release Products.
• Technical Knowledge Center, Field Notices
Includes information published by Alcatel Lucent Enterprise's Service and Support group.
• *Release Notes*
Includes critical Open Problem Reports, feature exceptions, and other important information on the features supported in the current release and any limitations to their support.

These user guides can be provided by contacting support or downloaded at Alcatel-Lucent Enterprise support website.

Telephone: 800.995.2696
Email: esd.support@alcatel-lucent.com
Support Web Site: http://service.esd.alcatel-lucent.com

1.8. **Before Calling Alcatel-Lucent's Technical Assistance Center**

Before calling Alcatel-Lucent's Technical Assistance Center (TAC), make sure that you have read through the appropriate section (or sections) and have completed the actions suggested for your system's problem.
Additionally, do the following and document the results so that the Alcatel-Lucent TAC can better assist you:
• Have a network diagram ready. Make sure that relevant information is listed, such as all IP addresses and their associated network masks.
• Have any information that you gathered while troubleshooting the issue to this point available to provide to the TAC engineer.
• If the problem appears to be with only a few-fewer than four-switches, capture the output from the "show tech-support" CLI command on these switches. (See Appendix C, "Technical Support Commands," for more information on **show tech-support** CLI commands.)
When calling Alcatel-Lucent TAC in order to troubleshoot or report a problem following information can be helpful to get a quick resolution:

- boot.cfg file (vcboot.cfg,vcsetup.cfg in case of Virtual Chassis)
- tech_support *.log & *.tar files, created by using:
  - show tech-support
  - show tech-support layer2
  - show tech-support layer3
  - show tech-support eng
- swlog, swlog.0, swlog.1 up to swlog.6 located in /flash/ and swlog_chassisX, swlog_chassisX.0, swlog_chassisX.1 up to swlog_chassisX.6 located in /flash/chassisX (where X is the active chassis number; e.g. 127.10.1.65)
- command.log file if present
- PMD files (Post Mortem Dump) if present (*pmd*) located in /flash/pmd and/or /flash/niX/pmd (where X is the active NI number; e.g. /flash/ni1/pmd)
- Captures of the following commands:
  - ls –lR (case sensitive)
  - show transceivers
  - show configuration status
  - show log swlog
  - show command-log (if command-log is enabled)
  - show user
  - show snmp statistics
  - show mac-learning
  - boardinfo
- Console output if captured during the issue
- SNMP Traps received by the NMS during the issue and any other

- Health graphs captured during that time
- Dial-in or remote access can also provide effective problem resolution.
- If a Virtual Chassis fail over to the secondary chassis happened because of this failure then include this information from both of the chassis.

```
6860-> debug show virtual-chassis connection
                          Address           Address
 Chas  MAC-Address        Local IP          Remote IP         Status
-----+------------------+-----------------+-----------------+-------------
 1     e8:e7:32:b3:34:51  127.10.2.65       127.10.1.65       Connected
 3     e8:e7:32:b3:36:9b  127.10.2.65       127.10.3.65       Connected
 4     e8:e7:32:b3:37:17  127.10.2.65       127.10.4.65       Connected
```

In SuperUser Mode the chassis are mounted as follows:

```
127.10.1.65:/flash/ on /mnt/chassis1_CMMA
127.10.2.65:/flash/ on /mnt/chassis2_CMMA
127.10.3.65:/flash/ on /mnt/chassis3_CMMA
127.10.4.65:/flash/ on /mnt/chassis4_CMMA

6860-> su
Entering maintenance shell. Type 'exit' when you are done.
SHASTA #-> ls /mnt/chassis2_CMMA/
811_555                      swlog_chassis2.1
bootflash                    swlog_chassis2.2
capManCmmTrace               swlog_chassis2.3
capManNiTrace                swlog_chassis2.4
certified                    swlog_chassis2.5
diags                        swlog_chassis2.6
eeprom                       swlog_chassis4
externalCPU                  swlog_chassis4.0
foss                         swlog_chassis4.1
fpga_name                    swlog_chassis4.2
hwinfo                       swlog_chassis4.3
issu                         system
lost+found                   tech_support.log
network                      u-boot.8.1.1.R01.462.tar.gz
pmd                          u-boot.8.1.1.R01.70.tar.gz
switch                       u-boot_copy
swlog                        vcboot.cfg.13.err
swlog_chassis1               vcboot.cfg.14.err
swlog_chassis2               working
swlog_chassis2.0
SHASTA #->
```

2. Troubleshooting the Switch System

In order to troubleshoot the system, a basic understanding of the operation of Chassis Management Modules (CMMs) and their interaction with Network Interface (NI) modules is required. Some concepts are covered in this chapter:

- Understanding of the “Diagnosing Switch Problems” chapter in the appropriate *OmniSwitch Switch Management Guide*.
- Understanding of the “Using Switch Logging” from the appropriate *OmniSwitch Network Configuration Guide* is highly recommended.

Summary of the commands in this chapter is listed here:

```
   show module status
   show powersupply
   show health
   show health all cpu
    show health configuration
    show swlog
   show log swlog
   top
   top -b -n 1 -m | head
  debug qos internal "chassis 1 slot 1 list 1 verbose"
  cat /proc/pktdrv
```

2.1. Introduction

The CMM is the Management Module of the switch. All of the critical operations of the switch including the monitoring is the responsibility of the CMM. CMM not only provides monitoring but also CMM synchronizes all of the NI for different operations.

2.2. Troubleshooting the System on OS6900/OS10K Switches

1. To troubleshoot system problems, the first step is to check the condition of all switch modules.

```
-> show module status
       Operational                Firmware
Slot      Status      Admin-Status   Rev           MAC
------+-------------+------------+---------+------------------
CMM-B   UP             POWER ON      2.0     e8:e7:32:9b:e2:6e
SLOT-1  UP             POWER ON      2.12    00:e0:b1:e4:c5:79
SLOT-2  UP             POWER ON      0.3     e8:e7:32:a5:d9:30
SLOT-5  UP             POWER ON      0.7     00:e0:b1:e4:ae:a1

-> show powersupply
           Total     Power    Input     PS
Slot  PS   Power     Used     Voltage   Type     Status   Location
---------+---------+--------+---------+--------+--------+-----------
      1     1200      16       120      AC       PWRSAVE     Internal
      2     1200       588       119      AC       UP          Internal
      3     1200       368       123      AC       UP          Internal
      4     1200       24        123      AC       UP          Internal
```

2. Check the CPU and memory status.

```
-> show health
CMM                  Current    1 Min    1 Hr   1 Day
Resources                         Avg      Avg    Avg
----------------------+---------+-------+-------+-------
CPU                       1        1       1       1
Memory                   22       22      22      22
```

Use the command “show health all cpu”, to examine the CPU on all modules.

```
-> show health all cpu
CPU                  Current    1 Min    1 Hr    1 Day
                              Avg      Avg     Avg
-------------------+----------+--------+-------+--------
Slot    1              11         11       11       10
Slot    2               8          8        7        7
Slot    5               5          5        5        5
```

3. Use the command “show log swlog” to check the output of the switch’s log files.

```
-> show log swlog
/flash/swlog_CMMB.7 not found!
/flash/swlog_CMMB.6 not found!
Displaying file contents for '/flash/swlog_CMMB.5'
Apr 28 20:26:48 (none) syslog.info syslogd started: BusyBox v1.19.3
Apr 28 20:26:48 (none) user.notice kernel: klogd started: BusyBox v1.19.3 (2013-03-18 16:33:36
PDT)
…..
```

2.3. Troubleshooting System on OS6860(E) Switches

If the switch is having problems the first place to look for is the CMM. All tasks are supervised by the CMM. Any inconsistencies between the CMM and the NI can cause problems.

1 The first step for troubleshooting problems with the switch is to look at the overall health of the switch.

Verify that all of the modules in the chassis are up and operational, using the command:

```
LAB-6860> show module status

                     Operational
 Chassis/Slot      Status      Admin-Status       MAC
--------------+-------------+------------+------------------
 1/CMM-A        UP             POWER ON      e8:e7:32:ae:78:11
 1/SLOT-1       UP             POWER ON      e8:e7:32:ae:78:18
```

The operational status can be DOWN while the power status is ON, indicating a possible software issue. For the CMM, the base chassis MAC address is displayed. For NI modules, the MAC address for the corresponding NI is displayed.

2 Verify the power supply (or supplies).

Check the power supply status, using the command:

```
sno-lab-r1-6860> show powersupply
```

```
             Total      PS
Chassis/PS       Power      Type      Status   Location
-----------+---------+--------+--------+-----------
1/1         150       AC       UP       Internal
Total   150
```

```
sno-lab-r1-6860> show powersupply 1
Module in slot PS-1
Model Name:                     PS-150AC,
Module Type:                    0x6040102,
Description:                    AC-PS,
Part Number:                    903400-90,
Hardware Revision:              A04,
Serial Number:                  1335000462,
Manufacture Date:               Sep  2 2013,
Operational Status:             UP,
Power Provision:                150W
```

Make sure that all the known good power supplies are operational.

3 Verify the CPU utilization.

The CPU utilization of CMM can be viewed by using the command:

```
LAB-6860> show health
CMM                  Current    1 Min    1 Hr   1 Day
Resources                        Avg      Avg    Avg
----------------------+---------+-------+-------+-------
CPU                        7        7       7       7
Memory                    61       61      56      56
```

The above command shows the memory, CPU statistics for current, 1 minimum average, 1 hour average and 1 hour maximum.

Check the threshold in health configuration, using command:

```
sno-lab-r1-6860> show health configuration
Rx Threshold                   = 80,
TxRx Threshold                 = 80,
CPU Threshold                  = 80,
Memory Threshold               = 80,
Sampling Interval (Secs)       = 10
```

All the values should be within the threshold. Any value above the threshold indicates abnormal behavior. The 1 hour average might be high if the switch was booted whithin the last hour but should normalize during the first hour of operation.

If none of the values are above the threshold, the next step is to attempt to isolate the problem to a particular NI, using command:

```
6860-> show health slot 1/1
Slot  1/ 1             Current    1 Min    1 Hr   1 Day
Resources                          Avg      Avg    Avg
----------------------+---------+-------+-------+-------
CPU                        8        7       7       6
Memory                    61       61      59      57
Receive                    0        0       0       0
Receive/Transmit           0        0       0       0
```

```
6860-> show health port 1/1/1
Port  1/ 1/ 1          Current    1 Min    1 Hr   1 Day
Resources                         Avg      Avg    Avg
----------------------+---------+-------+-------+-------
Receive                  0        0       0       0
Receive/Transmit         0        0       0       0
```

The above commands may help to narrow the problem to a particular NI or to the CMM. For more detail see, the section “Monitoring Switch Health” in the chapter titled “Diagnosing Switch Problems” in the appropriate *OmniSwitch Network Configuration Guide.*

4 Check the switch log.

One of the most important things to check is the switch log. The switch log contains the log events based on the settings of the log levels and applications configured to generate log events messages. Default settings of the switch log can be view using the command:

```
sno-lab-r1-6860> show swlog
Operational Status                 : Running,
File Size per file                 : 12500K bytes,
Log Device                         : console flash socket,
Log Device                         : ipaddr 192.168.2.131 remote command-log,
Syslog FacilityID                  : local7(23),
Hash Table entries age limit       : 60 seconds,
Switch Log Preamble                : Enabled,
Switch Log Debug                   : Disabled,
Switch Log Duplicate Detection     : Disabled,
Console Display Level              : info
```

By default, the chassis is set to log to flash and console. This can be changed and specific SYSLOG servers can be used to log the messages, please refer to the Switch Management Guide for further details. The default application trace level is ‘info’. Any error messages or informational messages would be logged in the switch log.

The switch log should be viewed to see if any error messages were generated by the switch. The command to use is:

```
sno-lab-r1-6860> show log swlog
/flash/swlog_chassis1.7 not found!
/flash/swlog_chassis1.6 not found!
/flash/swlog_chassis1.5 not found!
/flash/swlog_chassis1.4 not found!
/flash/swlog_chassis1.3 not found!
/flash/swlog_chassis1.2 not found!
Displaying file contents for '/flash/swlog_chassis1.1'
Jan  1 00:00:32 OS6860 syslogd started: BusyBox v1.19.3
Jan  1 00:00:32 OS6860 kernel: klogd started: BusyBox v1.19.3 (2014-05-14 02:47:33 PDT)
Jan  1 00:00:32 OS6860 kernel: [    0.000000] Booting Linux on physical CPU 0
Jan  1 00:00:32 OS6860 kernel: [    0.000000] Linux version 3.6.5
…….
```

If the log messages do not show enough information then they can be changed for specific applications to a higher log level or for all the applications running in the switch. For setting up different log levels in switch log, please refer to the “Using Switch Logging” chapter in the appropriate *OmniSwitch Network Configuration Guide.*

2.4. Advanced Troubleshooting

Troubleshooting High CPU utilization

1. First, identify which CPU is excessively high.

   CLI shell:
   6860->show health
   6860->show health all cpu
   Maintenance shell(AOS7 & 8 are Linux based, typing su in CLI will lead to Linux BASH shell):

   Top
   In the maintenance shell the "top" command is used to continuously monitor the tasks consuming CPU.

```
->top
Mem: 1172224K used, 849676K free, 0K shrd, 548K buff, 742380K cached
CPU:  0.0% usr  4.5% sys  0.0% nic 95.4% idle  0.0% io  0.0% irq  0.0% sirq
Load average: 0.23 0.19 0.15 2/237 3857
PID  PPID USER     STAT   VSZ %VSZ CPU %CPU COMMAND
3857  3854 root     R     2940  0.1   0  4.5 top
2090   813 root     S     301m 15.2   0  0.0 /bin/ipnid
2022   813 root     S     299m 15.1   1  0.0 /bin/bcmd -p
2081   813 root     S     296m 14.9   1  0.0 /bin/slNi
2105   813 root     S     277m 14.0   0  0.0 /bin/ipmsni
2102   813 root     S     277m 14.0   0  0.0 /bin/qosnid
1749   813 root     S     276m 14.0   0  0.0 /bin/qoscmmd
2069   813 root     S     268m 13.6   0  0.0 /bin/stpNi
2063   813 root     S     266m 13.4   0  0.0 /bin/lacpNi
```

2. The most common causes for high CPU utilization:

i. An abnormal process
   - A process goes into an infinite loop. This is probably a software issue.
   - A process is doing extensive calculations. It is possible that the network is not well scaled.
   - AOS is under a DoS attack.

ii. Abnormal traffic
   - Too many messages exchanged between AOS subsystems. Examples include: extensive logging, MAC learning, malfunctioning bus, HW interrupts.
   - Too many frames or packets are trapped to CPU

Proc	**Task**	**Proc**	**Task**
aaaCmm	AAA	mvrpNi	MVRP NI
agCmm	Access Guardian CMM	ntpd	NetworkTime Protocol
agNi	Access Guardian NI	ofcmmd	Open Flow CMM
appMonCmm	app-mon	ofnid	Open Flow NI
appMonNiStub	app-mon	pmCmm	Port mapping CMM
bcd	Broadcom driver	pmNi	Port mapping NI
bcmd	BCM	pmmCmm	Port Mirroring & Monitoring CMM
bfd	BFD CMM	pmmnid	Port Mirroring & Monitoring Ni

bfdni	BFD NI	portmgrcmm	Port manager CMM
capmanc	Capability Manager CMM	portmgrni	Port manager Ni
capmani	Capability Manager NI	qmrCmm	Quarantine Manager
dhcpsrv	DHCP	qosCmmd	QOS
dpiCmm	DPI	radCli	RADIUS client
dpiNi	DPI	remcfgMgr	Remote Config
eoamCmm	OAM	rmon	Remote Monitoring
eoamNi	OAM	saaCmm	Service Assurance Agent
erpCmm	ERP	sesmgrCmm	Session Manager
erpNi	ERP	sipCmm	SIP Snooping CMM
evbCmm	Edge Virtual Bridging CMM	sipNi	SIP Snooping Ni
evbNi	Edge Virtual Bridging NI	slCmm	SRC-LEARNING CMM
flashMgr	flash	slNi	SRC-LEARNING Ni
havlanCmm	vlan	slbcmmd	SLB
hmonCmm	Health CMM	stpCmm	STP CMM
hmonNi	Health NI	stpNi	STP NI
ipcmmd	IP	svcCmm	Service Manager (useb by SPB, MPLS/VPLS)
ipmscmm	IPMS CMM	tacClientCmm	TACACS
ipmsni	IPMS NI	trapmgr	TRAP
ipnid	ARP	udldCmm	UDLD CMM
iprm	IP Routing	udldNi	UDLD NI
ipsec6d	IPSEC	udpRelayCmmd	UDP Relay
isis	isis	udpRelayNi	DHCP relay/snooping
isisVc	Virtual-Chassis	vcmCmm	Virtual Chassis Manager CMM
lacpNi	LACP	vcmNi	Virtual Chassis Manager Ni
lagCmm	Link Agg	vcspCmm	Virtual Chassis Split Protection CMM
ldapClientCmm	LDAP	vcspNi	Virtual Chassis Split Protection Ni
lldpCmm	LLDP CMM	vfcm	Virtual Flow Controller CMM
lldpNi	LLDP NI	vfcn	Virtual Flow Controller Ni
loamNi	Link OAM	vmCmm	Vlan Manager CMM
lpCmm	Learned Port Security CMM	vmNi	Vlan Manager Ni
lpNi	Learned Port Security NI	vrrp	VRRP
mcipcd	Multi-Chassis IPC	vstkCmm	VLAN Stacking (Q-in-Q) CMM
mipgwd	Gateway software	vstkNi	VLAN Stacking (Q-in-Q) Ni
mvrpCMM	MVRP CMM	webMgtd	WEBVIEW

3. Identify the process causing high CPU usage

   Use the commands “top” and “ps” in the maintenance shell to find the process(es) consuming the most CPU and use knowledge of the network to determine whether or not the consumption is abnormal.

   Each process presents a task running in CPU. The meaning of process can be found in above figure.

4. Resolve the process causing high CPU

Find the specific chapter in this troubleshooting guide for the task causing high CPU. It is possible that the task needs to be restarted. Before restarting the task, please contact Alcatel-Lucent customer support.

5. Identify abnormal traffic

   It's worthwhile to verify how many packets are trapped by the CPU due to FFP rules (packets may be trapped to the CPU due to other reasons—see chapter Packet Driver for more details). The following commands will be helpful to locate abnormal traffic by showing the type and number of packets.

```
→debug qos internal "chassis 1 slot 1 list 1 verbose"
Entry U Slice CIDU CIDL MIDU MIDL TCAM   Count[+]  Green[+] Red[+]  NotGreen[+]
List 1: 41 entries set up
HgMcastARP(16) 0  14 - 3642 - -   3684     18656[18656] 0[0]    0[0]    0[0]
HgMcastARP(16) 0  15 -   -   - -   3940     18656[0]     0[0]    0[0]    0[0]
McastARP(17)   0  14 3591 - - -   3685         0[0]     0[0]    0[0]    0[0]
McastARP(17)   0  15  -   - - -   3941         0[0]     0[0]    0[0]    0[0]
ISIS_BPDU1(22) 0  14  - 3584 --   3639         0[0]     0[0]    0[0]    0[0]
ISIS_BPDU1(22) 0  15  -  - -  -   3895         0[0]     0[0]    0[0]    0[0]
ISIS_BPDU2(23) 0  14 3585- -  -   3640         0[0]     0[0]    0[0]    0[0]
```

2.5. **Memory Leak**

The following output shows memory utilization by process and can be helpful in determing which task is consuming memory, or if memory for a specific task is gradually increasing in the case of a memory leak.

```
SHASTA #-> top -b -n 1 -m | head
Mem total:2021900 anon:306672 map:639136 free:846800
 slab:35708 buf:572 cache:743544 dirty:4 write:0
Swap total:0 free:0
 PID   VSZ^VSZRW   RSS (SHR) DIRTY (SHR) STACK COMMAND
 2022  161m  135m 81244 10004 81228  9992   132 /bin/bcmd -p
1856 92640 83368  8108  7296  8108  7296   132 /bin/dhcpv6srv
 1868 91796 83240  8108  7276  8108  7276   132 /bin/dhcpsrv
 2102 40620 22688 29864 10196 29844 10180   132 /bin/qosnid
 2081 38956 20208 19468 10452 19444 10432   132 /bin/slNi
2105 32660 15884 22692  9888 22652  9856   132 /bin/ipmsni
```

2.6. **Packet Driver**

The packet driver is responsible for filtering and classifying all frames and packets trapped to the CPU.

```
SHASTA #-> cat /proc/pktdrv

Board Type          : 0x6062202
Tb                  : 4fa13fbd8f93d
Wakeups             : 26647195 26647194
Last Task Wakeup    :            0            0
Last Task Ran       : 4fa13fbd8ae6c   8903c213fe
Task Latency        :            3b fffb067507e965cd
Task Max Latency    : fffffffffffffc fffb067507e9658e
TX Chain Ints       : 24887887 24887884
TX Desc Ints        :        0        0
TX Timeout Ints     :        0        0
…..
Buffer States
Invalid             : 1
Free                : 1947
Classify            : 0
```

```
RX Dma              : 1755
TX Dma              : 0
…….. 
Task Info
Ipv4: [flags=1] [id 1] [ring size 100] [head 18] [tail 18]
Arp : [flags=1] [id 2] [ring size 256] [head 145] [tail 145]
Ip4t: [flags=1] [id 3] [ring size 100] [head 0] [tail 0]
……….. 
Classify Queues
Ipv4: Tx                  :  7645246  7645246
Ipv4: Rx Q Full Drops     :      135      135
Ipv4: Rx                  :  7640318  7640318
……….. 
```

2.7. The second column of output provides changes since the previous execution of the command.Logs

Besides the swlog, the kernel log in /var/log/.

3. Troubleshooting Virtual Chassis

3.1. **Introduction**

A Virtual Chassis is a group of switches managed through a single management IP address that operates as a single bridge and router. It provides both node level and link level redundancy for layer 2 and layer 3 services and protocols acting as a single device. The use of a virtual chassis provides node level redundancy without the need to use redundancy protocols such as STP and VRRP between the edge and the aggregation/core layer.

Some of the key benefits provided by Virtual Chassis are:

- A single, simplified configuration to maintain
- Optimized bandwidth usage between the access layer and core
- Active-Active multi-homed link aggregation
- Provides predictable and consistent convergence with redundant links to the two switches
- Allows for exclusion of spanning-tree and other redundancy protocols like VRRP between the access layer and the core
- A Virtual Chassis appears as single router or bridge with support for all protocols
- A Virtual Chassis can be upgraded using ISSU to minimize network impact

Summary of the commands in this chapter is listed here:

```
show virtual-chassis topology
show virtual-chassis consistency
show virtual-chassis vf-link
show virtual-chassis vf-link member-port
show virtual-chassis chassis-reset-list
show running-directory
show interfaces status
show log swlog | grep vcmCmm
debug show virtual-chassis topology
debug show virtual-chassis status
cat vc_debug.txt
```

Master Chassis Election :

- The learning window is 30 seconds after VFL comes up
- Master chassis election is based on:
- Highest chassis priority

```
-> virtual-chassis configured-chassis-priority 100
```

The highest number configured-chassis-priority will become the Master chassis.
Without setting this value the lowest chassis identifier becomes the key value used to determine which switch will become the Master.

- Longest Chassis uptime
- Lowest Chassis ID
- Lowest Chassis MAC Address

In OS6860/OS6860E auto VC, once the Master is elected the chassis connecting the lower VFL port of the master is assigned Chassis 2 and the value increments for each additional chassis up to chassis 8.

Virtual Chassis - Boot-Up

The Master chassis contains the vcboot.cfg file that details the configuration for the entire virtual chassis. All the switches (i.e. the one that will eventually become the Master and the ones that will become Slaves) contain a vcsetup.cfg file that allows them to establish an initial connection over a VFL to all the other neighboring switches.

1. Upon boot-up, a switch will read its local vcsetup.cfg file and attempt to connect to the other neighbor switches
2. Upon connection, the switches will exchange the parameters configured in their local vcsetup.cfg files
3. As a result of this exchange, they will discover the topology, elect a Master based on criteria described in the next section, start periodic health checks over the VFL, and synchronize their configuration as defined within the vcboot.cfg configuration file
4. All Slaves, if they do not have a local copy of vcboot.cfg, or if their local copy does not match the copy found on the Master, will download the vcboot.cfg from the Master chassis and reboot using this copy of vcboot.cfg as its configuration file

Startup Error Mode

If a switch is unable to successfully come up in virtual chassis mode, it enters a special fallback mode called start up error mode. A switch moves to start up error mode if either one of the following conditions occur:

- The vcsetup.cfg and vcboot.cfg configuration files are present in the running directory, but no valid advanced license is installed on the switch.
- The vcsetup.cfg file is corrupted or edited in such a way that it is unable to read a valid chassis identifier in the appropriate range.

A switch start up error mode will keep all of its front-panel user ports, including the virtual-fabric links member ports disabled. This mode can be identified on the switch by using the show virtual-chassis topology command. The chassis role will display Inconsistent, whereas the chassis status will show either one of the following values:

- Invalid-Chassis-Id: The chassis is not operational in virtual chassis mode because no valid chassis identifier has been found in the configuration. Typically this means that the vcsetup.cfg file is corrupted, empty or contains an invalid (e.g. out of range) chassis identifier.

- Invalid-License: The chassis is not operational in virtual chassis mode because no valid advanced license has been found.

3.2. **Basic Troubleshooting**

This command is used to provide a detailed status of the virtual chassis topology.

```
OS6860-> show virtual-chassis topology
Local Chassis: 1
 Oper                                   Config   Oper
 Chas  Role         Status              Chas ID  Pri   Group  MAC-Address
-----+------------+-------------------+--------+-----+------+------------------
 1     Master       Running             1        100   0      e8:e7:32:b3:3c:3b
 2     Slave        Running             2        100   0      e8:e7:32:b3:49:11
```

This command is used to provide a detailed status of the parameters taken into account to determine the consistency of a group of switches participating in the virtual chassis topology.

```
OS6860-> show virtual-chassis consistency
Legend: * - denotes mandatory consistency which will affect chassis status
        licenses-info - A: Advanced; B: Data Center;

       Config           Oper                    Oper     Config
       Chas             Chas    Chas   Hello    Control  Control
 Chas* ID     Status    Type*   Group* Interv   Vlan*    Vlan     License*
------+------+---------+-------+------+-------+--------+--------+----------
 1     1      OK        OS6860  0      15       4094     4094
 2     2      OK        OS6860  0      15       4094     4094
```

A more detailed version of this output is available after adding the chassis-id option

```
OS6860-> show virtual-chassis chassis-id 1 consistency
Legend: * - denotes mandatory consistency which will affect chassis status
        licenses-info - A: Advanced; B: Data Center;

                       Given       Master
Consistency            Chassis     Chassis     Status
---------------------+-----------+-----------+---------
Chassis-ID*            1           1           OK
Config-Chassis-ID      1           1           OK
Chassis-Type*          OS6860      OS6860      OK
License*                                       OK
Chassis-Group*         0           0           OK
Hello-Interval         15          15          OK
Oper-Control-Vlan*     4094        4094        OK
Config-Control-Vlan    4094        4094        OK
```

Displays a summary of the configured and operational parameters related to the virtual fabric links on the virtual chassis topology

```
OS6860-> show virtual-chassis vf-link
                              Primary   Config  Active  Def       Speed
 Chassis/VFLink ID  Oper       Port      Port    Port    Vlan      Type
-------------------+----------+---------+-------+-------+---------+-----------
 1/0                Up         1/1/30    2       2       1         21G
 2/0                Up         2/1/30    2       2       1         21G
```

And per link

```
OS6860-> show virtual-chassis vf-link member-port
 Chassis/VFLink ID  Chassis/Slot/Port  Oper       Is Primary
-------------------+------------------+----------+-------------
 1/0                1/1/29             Up         No
 1/0                1/1/30             Up         Yes
 2/0                2/1/29             Up         No
 2/0                2/1/30             Up         Yes
```

This command displays the list of all chassis that must be reset along with a specified chassis in order to prevent a virtual chassis topology split

```
OS6860-> show virtual-chassis chassis-reset-list
 Chas  Chassis reset list
-----+--------------------
 1     1,
 2     2,
```

Display Virtual Chassis logs from SWLOG

```
-> show log swlog | grep vcmCmm
```

3.3. **Advanced Troubleshooting**

The below command lets the user know whether the unit has reached the ready state or not.

```
OS6860-> debug show virtual-chassis topology
Local Chassis: 1
 Oper                                  Config   Oper                                    System
 Chas  Role         Status             Chas ID  Pri   Group  MAC-Address       Ready
-----+------------+-------------------+--------+-----+------+------------------+-------
 1     Master       Running            1        100   0      e8:e7:32:b3:3c:3b  Yes
 2     Slave        Running            2        100   0      e8:e7:32:b3:49:11  Yes
```

Warning: "RCD Operational Status" is "Up" even in case there is a duplicated chassis group ID in the same network!

The below command lets the user know on which state the VC has failed :

```
OS6860-> debug show virtual-chassis status

 ID  Level  Parameter                     Value            Timestamp   Status
----+------+-----------------------------+----------------+-----------+---------
 0   L0     Chassis Identifier            1                01:03:57    OK
 1   L0     Designated NI Module          1                01:03:57    OK
 2   L0     Designated NI Module (@L5)    1                00:40:45    OK
 3   L0     License Configured            Yes              01:03:57    OK
 4   L0     License Configured (@L5)      Yes              00:40:45    OK
 5   L0     VFL Links Configured          2                01:03:57    OK
 6   L0     VFL Links Configured (@L5)    2                00:40:45    OK
 7   L0     VFL Ports Configured          2                01:03:57    NOK_08
 8   L0     VFL Ports Configured (@L5)    2                00:40:45    OK
 11  L0     Chassis Ready Received        Yes              00:40:38    OK
 12  L1     VFL Intf Oper Status          Down             01:03:57    NOK_09
 13  L1     VFL Intf Oper Status (@L5)    Down             00:40:45    NOK_09
 14  L2     VFL LACP Status               Down             01:03:57    NOK_14
 15  L2     VFL LACP Status (@L5)         Down             00:40:45    NOK_14
 16  L2     VFL LACP Up -> Down           3                00:46:05    INFO_04
 17  L2     VFL LACP Down -> Up          4                00:45:56    INFO_03
 18  L3     VCM Protocol Role (@L5)      Slave            00:40:45    OK
 19  L3     VCM Protocol Role            Master           01:03:57    OK
 20  L3     VCM Protocol Status (@L5)    Running          00:40:45    OK
 21  L3     VCM Protocol Status          Running          01:03:57    OK
 24  L4     VCM Connection               Up               01:03:57    OK
 25  L4     VCM Connection (@L5)         Up               00:40:45    OK
 26  L5     VCM Synchronization         Multi-node        01:03:57    OK
 27  L6     Chassis Sup Connection      Up                00:46:00    OK
 28  L6     Remote Flash Mounted         Yes              00:46:26    OK
 29  L6     Image and Config Checked     Yes              00:41:01    OK
 30  L6     VC Takeover Sent             Yes              00:42:55    OK
 31  L7     VC Takeover Acknowledged     Yes              00:42:58    OK
 32  L8     System Ready Received        Yes              00:41:06    OK
 33  L8     RCD Operational Status       N/A              N/A         N/A
 34  L8     RCD IP Address               N/A              N/A         N/A

Error/Information Codes Detected:
--------------------------------------------------------------------------------
NOK_08
    There are no virtual-fabric member ports configured on this switch.
    If there are multiple virtual-fabric links configured, we must have
    at least one member port configured or assigned to each of the
    virtual-fabric links.
    Troubleshooting Tips:
    -> show virtual-chassis vf-link member-port | grep "<local-chassis-id>/"

NOK_09
    There are no virtual-fabric member interfaces operationally up.
    If there are multiple virtual-fabric links configured, we must have
    at least one member port interface up on each virtual-fabric link.
    Troubleshooting Tips:
    -> show virtual-chassis vf-link member-port | grep "<local-chassis-id>/"
    -> show interfaces port <chassis>/<slot>/<port> status

NOK_09
    There are no virtual-fabric member interfaces operationally up.
    If there are multiple virtual-fabric links configured, we must have
    at least one member port interface up on each virtual-fabric link.
    Troubleshooting Tips:
    -> show virtual-chassis vf-link member-port | grep "<local-chassis-id>/"
    -> show interfaces port <chassis>/<slot>/<port> status

NOK_14
    The virtual-fabric links configured on this switch are not operationally up.
    If there are multiple links configured, all of them must be operationally up
    in order for this parameter to be reported as OK.
    Troubleshooting Tips:
    -> show virtual-chassis vf-link | grep "<local-chassis-id>/"

NOK_14
    The virtual-fabric links configured on this switch are not operationally up.
    If there are multiple links configured, all of them must be operationally up
    in order for this parameter to be reported as OK.
    Troubleshooting Tips:
    -> show virtual-chassis vf-link | grep "<local-chassis-id>/"

INFO_04
    This parameter provides the counter of how many times the
    virtual-fabric link operational status has transitioned from
    up to down since the switch was started. Depending on the specific
    operations you are performing on the system, this counter may
    increase. Under normal conditions this counter should ideally
    be one unit smaller than the counter for the opposite transitions
    from operational down to up

INFO_03
    This parameter provides the counter of how many times the
    virtual-fabric link operational status has transitioned from
    down to up since the switch was started. Depending on the specific
    operations you are performing on the system, this counter may
    increase. Under normal conditions this counter should ideally
    be one unit greater than the counter for the opposite transitions
    from operational up to down
```

Troubleshooting in maintenance shell :

Port status in the hardware. To check if the ports which are mapped to VFL are in forwarding state, list device ports providing device port numbers. When a port is configured as a VFL, it is renamed from xeM to hgN, where M and N are numbers.

```
OS6860-> su
Entering maintenance shell. Type 'exit' when you are done.
SHASTA #-> bShell

Entering character mode
Escape character is '^]'.

Broadcom Command Monitor: Copyright (c) 1998-2010 Broadcom Corporation
Release: sdk-6.3.4 built  ()
From @:7.1.1.R01
Platform: unknown
OS: Unix (Posix)
BCM.0> ps
            ena/    speed/ link auto     STP                  lrn  inter   max  loop
      port  link    duplex scan neg?    state   pause  discrd ops   face frame  back
       ge0  down      -      SW  Yes     Block          None   FA  SGMII  9216
       ge1  down      -      SW  Yes     Block          None   FA  SGMII  9216
       ge2  down      -      SW  Yes     Block          None   FA  SGMII  9216
.
.
.
.
.
       xe1  !ena   10G  FD None  No      Block          None   FA    XFI  9216
       xe2  !ena   10G  FD None  No      Block          None   FA    XFI  9216
       xe3  !ena   10G  FD None  No      Block          None   FA    XFI  9216
       hg0  up     21G  FD   HW  Yes  Forward          None   FA  XGMII 16360
       hg1  up     21G  FD   HW  Yes  Forward          None   FA  XGMII 16360
```

To collect debug traces and data either to a file or, if no file is specified to the standard output use the command "debug dump virtual-chassis type {trace | data | all} chassis-id <chassis-id> slot {<slot-id> | all} [output-file <filepath/filename>]". Example:
The logs provided complete log about the VC states,VFL linkstates, VC Event-traces,etc.

```
OS6860->
flash/vc_debug.txtp virtual-chassis chassis-id 1 type all slot all output-file /
Please wait.......................................................

Output File        : /flash/vc_debug.txt successfuly created (size 670822 bytes)

OS6860-> cat vc_debug.txt
       ack,         : Request acknowledgment {1 = true, 0 = false}
       dump)        : Request packet dump {1 = true, 0 = false}
> vcm_debug_socket
       socket,      : Socket < 0 => debug reactor all sockets only
       debugAllSock,: 1/0 enable/disable all socket debug
       debugReac    : 1/0 enable/disable reactor debug

Public msg traces and statistics
---------------------------------------------------------------------------------
> vcm_show_public_trace (start, count, detail)
      show range of public messages in the trace buffer. Use start = 0 and count = 0
      to show all msgs in the trace buffer
> vcm_clean_public_trace ()
      clean up IPC message trace buffer
. . . . . . .
```

3.4. **Scenarios :**

To add a switch into an existing Virtual Chassis:

*** The failed state:
Prior to connecting the VFL cable, connect the console cable to the new non-configured unit. Determine what directory the chassis is running from with the 'show running-directory command'. It is possible it is currently running at the certified directory.

```
-> show running-directory

CONFIGURATION STATUS
  Running CMM                : MASTER-PRIMARY,
  CMM Mode                    : VIRTUAL-CHASSIS MONO CMM,
  Current CMM Slot          : CHASSIS-1 A,
  Running configuration    : CERTIFIED,
  Certify/Restore Status   : CERTIFIED
SYNCHRONIZATION STATUS
  Running Configuration    : NOT SYNCHRONIZED
```

Prepare the directory from which the switch has to boot from:

```
-> pwd
/flash
-> mkdir vc_dir
-> cd working

-> ls
Uos.img       boot.md5      software.lsm  vcboot.cfg    vcsetup.cfg
-> cp Uos.img /flash/vc_dir
-> cp vcsetup.cfg /flash/vc_dir


-> reload from vc_dir no rollback-timeout
Confirm Activate (Y/N) : Y
This operation will verify and copy images before reloading.
It may take several minutes to complete....

Fri Jan  3 23:43:05 : ChassisSupervisor vcReloadMgr info message:
+++ vcReloadMgrReloadVC: starting reload sequence for image vc_dir
```

After the switch is rebooted, the switch is now running from the vc_dir directory:

```
-> show running-directory

CONFIGURATION STATUS
  Running CMM              : SLAVE-PRIMARY,
  CMM Mode                 : VIRTUAL-CHASSIS MONO CMM,
  Current CMM Slot         : CHASSIS-1 A,
  Running configuration    : vc_dir,
  Certify/Restore Status   : CERTIFY NEEDED
SYNCHRONIZATION STATUS
  Running Configuration    : SYNCHRONIZED
```

For example, you want to add virtual Chassis 7 into a stack of 6, you can issue the follow command to the new unit.

```
->virtual-chassis chassis-id 1 configured-chassis-id 7

-> write memory

File /flash/vc_dir/vcsetup.cfg replaced.

File /flash/vc_dir/vcboot.cfg replaced.


-> copy running certified flash-synchro

-> reload from vc_dir no rollback-timeout
```

After the switch is completely rebooted, you now should see the unit is now unit 7 on the digital display.(OS6860 allows up to 8 chassis, OS6900 allows up to 6 chassis)

Now you can power off the unit, and then connect this unit to the main stack by using the VFL cables. Then power up the newly inserted unit. The newly inserted unit will boot and sync-up the software and configuration file with the Master unit of the Virtual Chassis.

To reload a specific switch in a virtual chassis (e.g. To reboot only chassis-ID 4):

```
OS6860-10.255.13.68-> reload chassis-id 4 from issu no rollback-timeout
Confirm Activate (Y/N) : Y
```

To look at what directory the switch is currently running:

```
OS6860-10.255.13.68-> show running-directory

CONFIGURATION STATUS
 Running CMM              : MASTER-PRIMARY,
 CMM Mode                 : VIRTUAL-CHASSIS MONO CMM,
 Current CMM Slot         : CHASSIS-1 A,
 Running configuration    : vc_dir,
 Certify/Restore Status   : CERTIFIED
SYNCHRONIZATION STATUS
 Flash Between CMMs       : SYNCHRONIZED,
 Running Configuration    : SYNCHRONIZED
```

In this example, the switch is running from **vc_dir, directory.**

And to check the file size,

```
OS6860-10.255.13.68-> cd /flash/vc_dir
OS6860-10.255.13.68-> ls -l
total 206668
-rw-r--r--    1 admin    user     211389068 Feb  9 23:55 Uos.img
-rw-------    1 root     root            40 Mar  4 06:24 boot.md5
-rw-r--r--    1 admin    user          2723 Mar  4 06:10 vcboot.cfg
-rw-r--r--    1 admin    user           216 Mar  4 06:10 vcsetup.cfg
```

- The Uos.img is AOS image file that boots up the switch.
- boot.md5 is a binary file that self generated during boot up.
- vcboot.cfg is the switch configuration file.
- vcsetup.cfg is the virtual chassis configuration file, where VFL ports are defined.

Note that vcboot.cfg is the same for both Master virtual chassis and Slave virtual chassis, but the vcsetup.cfg is different for each virtual-chassis element.

To check the entire VC interface adminstrative status:

```
OS6860-10.255.13.68-> show interfaces status
Chas/                   DETECTED-VALUES         CONFIGURED-VALUES
Slot/    Admin  Auto  Speed   Duplex  Pause   Speed   Duplex  Pause  Link
Port     Status Nego  (Mbps)                  (Mbps)                 Trap  EEE
---------+------+----+--------+------+-------+--------+------+-------+-----+---
 1/1/1      en    en    1000   Full     -      Auto    Auto     -     dis  dis
 1/1/2      en    en     -      -       -      Auto    Auto     -     dis  dis
 1/1/3      en    en     -      -       -      Auto    Auto     -     dis  dis
 1/1/4      en    en     -      -       -      Auto    Auto     -     dis  dis
 1/1/5      en    en     -      -       -      Auto    Auto     -     dis  dis
 1/1/6      en    en     -      -       -      Auto    Auto     -     dis  dis
  …….. 
```

To display all the VFL ports on the VC:

```
OS6860-10.255.13.68-> show virtual-chassis vf-link member-port
 Chassis/VFLink ID  Chassis/Slot/Port  Oper       Is Primary
-------------------+------------------+----------+-------------
 1/0                1/1/29             Up         Yes
 1/1                1/1/30             Up         Yes
 2/0                2/1/29             Up         Yes
 2/1                2/1/30             Up         Yes
 3/0                3/1/53             Up         Yes
 3/1                3/1/54             Up         Yes
 4/0                4/1/29             Up         Yes
 4/1                4/1/30             Up         Yes
 5/0                5/1/29             Up         Yes
 5/1                5/1/30             Up         Yes
 6/0                6/1/29             Up         Yes
 6/1                6/1/30             Up         Yes
 7/0                7/1/53             Up         Yes
```

Troubleshoot flapping VFL port(s).

To temporarily disable a VFL-link for troubleshooting purposes:

```
OS6860-10.255.13.68-> debug interfaces port 1/1/29 admin-state disable 1/1/29 Auto
Thu Mar 13 01:41:08 : vcmCmm port_mgr info message:
+++ CMM:vcmCMM_client_rx_pm@1801: VFL link 2/0 down (last 2/1/29) [L2]
OS6860-10.255.13.68-> show virtual-chassis vf-link member-port
 Chassis/VFLink ID  Chassis/Slot/Port  Oper       Is Primary
-------------------+------------------+----------+-------------
 1/0                1/1/29             Down       No
 1/1                1/1/30             Up         Yes
 2/0                2/1/29             Down       No
 2/1                2/1/30             Up         Yes
 3/0                3/1/53             Up         Yes
 3/1                3/1/54             Up         Yes
 4/0                4/1/29             Up         Yes
 4/1                4/1/30             Up         Yes
 5/0                5/1/29             Up         Yes
 5/1                5/1/30             Up         Yes
 6/0                6/1/29             Up         Yes
 6/1                6/1/30             Up         Yes
 7/0                7/1/53             Up         Yes
```

To check consistency of a virtual chassis:

```
OS6860-10.255.13.68-> show virtual-chassis consistency
     Config           Oper                  Oper     Config
       Chas             Chas    Chas   Hello   Control  Control
```

```
 Chas* ID     Status    Type*   Group* Interv  Vlan*    Vlan     License*
------+------+---------+-------+------+-------+--------+--------+----------
 1     1      OK        OS6860  0      15      4094     4094
 2     2      OK        OS6860  0      15      4094     4094
 3     3      OK        OS6860  0      15      4094     4094
 4     4      OK        OS6860  0      15      4094     4094
 5     5      OK        OS6860  0      15      4094     4094
 6     6      OK        OS6860  0      15      4094     4094
```

To check IPC (inter-switch) connectivity:

```
Ping 127.10.2.65      // CMM-A on Chassis 2
Ping 127.10.2.1       // NI-1 on Chassis 2
```

To check all the inter-switch connectivity.

To lookup a MAC address and see if it is learned on the hardware

To display all the MACs learned on the hardware:

```
OS6860-10.255.13.68-> su
Entering maintenance shell. Type 'exit' when you are done.
SHASTA #-> bShell
BCM.0> l2 show
mac=e8:e7:32:b3:45:cd vlan=100 modid=0 port=0/cpu0 Static Hit CPU
mac=00:0a:1e:22:07:f8 vlan=4094 modid=24 port=0   Hit
mac=00:0a:1e:22:02:51 vlan=4094 modid=4 port=0   Hit
mac=00:0a:1e:22:06:f8 vlan=4094 modid=20 port=0   Hit
mac=00:0a:1e:22:03:51 vlan=4094 modid=8 port=0   Hit
……
```

To verify inter-process communication between switches.

```
SHASTA #-> ping -c 10 -s 400 -f 127.10.2.65
PING 127.10.2.65 (127.10.2.65) 400(428) bytes of data.

--- 127.10.2.65 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 24ms
rtt min/avg/max/mdev = 0.556/2.412/9.297/2.984 ms, ipg/ewma 2.759/3.191 ms
```

To check if the VCM process was able to establish inter-chassis ipc connections

```
OS6860-10.255.13.68-> debug show virtual-chassis connection
                          Address           Address
 Chas  MAC-Address        Local IP          Remote IP         Status
-----+------------------+-----------------+-----------------+-------------
 2     e8:e7:32:b3:3c:3b  127.10.1.65       127.10.2.65       Connected
 3     e8:e7:32:b3:34:cd  127.10.1.65       127.10.3.65       Connected
 4     e8:e7:32:b3:49:11  127.10.1.65       127.10.4.65       Connected
 5     e8:e7:32:ab:21:b9  127.10.1.65       127.10.5.65       Connected
 6     e8:e7:32:b3:3b:ef  127.10.1.65       127.10.6.65       Connected
 7     e8:e7:32:ab:1c:d3  127.10.1.65       127.10.7.65       Connected
```

The above command will also show the internal ip address for every switch in the virtual-chassis.

Remote Chassis Detection (RCD) occurs every 1 seconds between the switches in the virtual chassis when the EMP ports are configured.

```
OS6860-10.255.13.68-> show virtual-chassis vf-link member-port
 Chassis/VFLink ID  Chassis/Slot/Port  Oper       Is Primary
```

```
-------------------+------------------+----------+-------------
 1/0                1/1/29             Up         Yes
 1/1                1/1/30             Up         Yes
 2/0                2/1/29             Up         Yes
 2/1                2/1/30             Up         Yes
 3/0                3/1/53             Up         Yes
 3/1                3/1/54             Up         Yes
 4/0                4/1/29             Up         Yes
 4/1                4/1/30             Up         Yes
 5/0                5/1/29             Up         Yes
 5/1                5/1/30             Up         Yes
 6/0                6/1/29             Up         Yes
 6/1                6/1/30             Up         Yes
 7/0                7/1/53             Up         Yes
 7/1                7/1/54             Up         Yes
```

4. Troubleshooting Switched Ethernet Connectivity

Summary of the commands in this chapter is listed here:

show interfaces
show vlan members
show mac-learning

4.1. **Verify Physical Layer Connectivity**

Verify that there is valid link light along the entire data path between the devices that can not switch to each other. Make sure to include all inter-switch links. Verify LEDs on all involved CMMs and NIs are solid OK1, blinking OK2 for OS10K or Solid OK for OS6900.
Use the **show interfaces** command to verify that operational status is up, speed and duplex are correct and match the device on the other side of the connection.

```
-> show interfaces 1/1/2
Chassis/Slot/Port  1/1/2  :
 Operational Status     : up,
 Last Time Link Changed : Thu Jan  9 04:29:54 2014,
 Number of Status Change: 1,
 Type                   : Ethernet,
 SFP/XFP                : N/A,
 EPP                    : Disabled,
 Link-Quality           : N/A,
 MAC address            : e8:e7:32:ab:1c:5f,
 BandWidth (Megabits)   :     1000,            Duplex          : Full,
 Autonegotiation        :   1  [ 1000-F 100-F 100-H 10-F 10-H ],
 Long Frame Size(Bytes) : 9216,
 Rx              :
 Bytes Received  :          20958326859, Unicast Frames :            102006483,
 Broadcast Frames:             28703682, M-cast Frames  :             99646487,
 UnderSize Frames:                    0, OverSize Frames:                    0,
 Lost Frames     :                    0, Error Frames   :                    0,
 CRC Error Frames:                    0, Alignments Err :                    0,
 Tx              :
 Bytes Xmitted   :          20345910300, Unicast Frames :             99450219,
 Broadcast Frames:             30870278, M-cast Frames  :            100511377,
 UnderSize Frames:                    0, OverSize Frames:                    0,
 Lost Frames     :                    0, Collided Frames:                    0,
 Error Frames    :                    0
```

If the port reports operational status down, verify the physical link, but also verify the necessary NIs and CMM are receiving power and are up and operational. Use the **show chassis** command and the **show cmm** command to verify this.

4.2. **Verify Current Running Configuration**

If the physical layer is validated, the next step is to verify the configuration. Use the **show configuration snapshot all** command to display the current running configuration. Use this command to verify the ports involved are in thecorrect VLAN. Also review the output of the command to verify there is nothing explicit in the configuration that would cause the problem, such as a deny ACL that could be found under the QoS subsection.

Additionally, verify the ports are in the correct VLAN and in a spanning tree forwarding state instead of blocking by using the **show vlan member** command.

```
-> show vlan members
  vlan       port         type         status
--------+------------+------------+--------------
  1         1/1/1        default    inactive
  1         1/1/2        default    forwarding
  1         1/1/3        default    forwarding
  1         1/1/4        default    inactive
  1         1/1/5        default    inactive
  1         1/1/6        default    inactive
```

If ports that should be in forwarding are in blocking, or vice versa, please consult the chapter for troubleshooting spanning tree.

4.3. **Verify Source Learning**

If the configuration looks correct, source learning should be examined. If connectivity exists but is slow or intermittent, source learning could be the root cause since data packets would be flooded if the MAC address(es) are not being learned. However, if there is no packet throughput between the devices the problem is likely not due to a source learning problem. To verify that the MAC addresses are being learned correctly use the **show mac-learning** command. Verify that the correct MAC address is being learned on the correct port, in the correct VLAN.

```
-> show mac-learning
Legend: Mac Address: * = address not valid,

        Mac Address: & = duplicate static address,

   Domain       Vlan/SrvcId/ISId         Mac Address           Type          Operation          Interface
------------+---------------------+-------------------+------------------+-------------+---------------------
      VLAN                       1   00:00:5e:00:01:02           dynamic     bridging            1/1/2
      VLAN                       1   00:1a:1e:00:5b:60           dynamic     bridging            1/1/2
      VLAN                       1   00:d0:95:e0:78:98           dynamic     bridging            1/1/2
```

In most case, the output of show mac-learning is too long to find the one needed. We can use parameter “grep” to find the MAC address we need, “show mac-learning | grep xx:yy –B 5”
With xx:yy as the last 4 of the mac address of interest. Then –B 5 prints a the first few lines (headings) before the lines with matching content.

4.4. **Verify Switch Health**

If source learning appears to be working incorrectly, verify the health of the switch with the **show health**, and **show health slot** commands. Any values that have reached or exceeded their configured threshold could cause forwarding problems on the switch.

```
-> show health
CMM                   Current    1 Min    1 Hr   1 Day
Resources                          Avg      Avg    Avg
----------------------+---------+-------+-------+-------
CPU                        9         6       6       6
Memory                    56        56      56      55
```

4.5. **Verify ARP**

If everything checked appears to be valid, verify that this is not an ARP problem. On the end stations involved, enter a static MAC address for the device it is trying to communicate with. If connectivity is restored, please see Chapter 6 Troubleshooting ARP

5. Troubleshooting Source Learning/Layer 2 Forwarding

Introduction

When a packet first arrives on NI source learning examines the packet and tries to classify the packet to join its correct VLAN. If a port is statically defined in a VLAN, the MAC address is classified in the default VLAN. Alternatively, if UNP is being used the MAC address is classified into the correct VLAN based on the rules defined.

As soon as the MAC address is classified in a VLAN, an entry is made in source address pseudo-CAM associating the MAC address with the VLAN ID and the source port. This source address is then relayed to the CMM for management purposes.

If an entry already exists in MAC address database with the same VLAN ID and the same source port number then no new entry is made. If the VLAN ID or the source port is different from the existing entry in the MAC address database then the previous entry is aged out and a new entry is made in the MAC address database. This process of adding a MAC address in the MAC address database is known as source learning.

A MAC address can be denied to learn on a port based on different policies configured through QoS or Learned Port Security. A MAC address may be learned in a wrong VLAN based on the policies defined for the port.

Summary of the commands in this chapter is listed here:

---

show mac-learning summary
show mac-learning mac-address <MAC address>
show mac-learning port <slot/port>
show mac-learning aging-time
show interfaces | grep Number
show interfaces | grep Last
show spantree vlan <vlan-id>
debug $(pidof stpNi) "call stpni_printStats(1,1)"
l2 show

5.1. Basic Troubleshooting

In order to troubleshoot a source learning problem the first step is to verify that the physical link is up and the port has correctly auto-negotiated with the end-station.

The next thing is to verify that the port is a member of the right VLAN, if a port is statically configured for a VLAN, or the UNP policies are correctly defined. The workstation configuration should also be verified.

Check the current MAC table size by using the below command to understand the number of MAC addresses learned on a switch.

```
6860-> show mac-learning summary
Mac Address Table Summary:

   Domain       Static     Static-Multicast       Bmac        Dynamic
```

```
------------+------------+------------------+------------+------------
    VLAN             0                  0            0           2841
    VPLS             0                  0            0           0
     SPB             0                  0            0           0
     EVB             0                  0            0           0

 Total MAC Address In Use  = 2841
```

If the MAC address table size is large, then use the additional options for looking at the specific MAC address in question.

The **mac-address** key-word can be used to search for the exact MAC address in question.

Note: There are scenarios when the same MAC address will be learned on different VLANs. This is most common when you have two switches connected together with multiple VLANs configured between them.

```
-> show mac-learning mac-address e8:e7:32:00:ef:a2
Legend: Mac Address: * = address not valid,

        Mac Address: & = duplicate static address,

   Domain     Vlan/SrvcId/ISId        Mac Address        Type          Operation    Interface
-----------+--------------------+-------------------+---------------+------------+----------
      VLAN                 1    e8:e7:32:00:ef:a2         dynamic     bridging      2/1/7
      VLAN                 10   e8:e7:32:00:ef:a2         dynamic     bridging      2/1/7
      VLAN                 20   e8:e7:32:00:ef:a2         dynamic     bridging      2/1/7
      VLAN                 30   e8:e7:32:00:ef:a2         dynamic     bridging      2/1/7

 Total number of Valid MAC addresses above = 4
```

The **port** or **linkagg** option can also be used to see the mac-addresses learned on a port or linkagg.

```
-> show mac-learning port 2/1/7
Legend: Mac Address: * = address not valid,

        Mac Address: & = duplicate static address,

   Domain     Vlan/SrvcId/ISId        Mac Address        Type          Operation    Interface
-----------+--------------------+-------------------+---------------+------------+----------
      VLAN                 1    e8:e7:32:00:ef:a2         dynamic     bridging      2/1/7
      VLAN                 10   e8:e7:32:00:ef:a2         dynamic     bridging      2/1/7
      VLAN                 20   e8:e7:32:00:ef:a2         dynamic     bridging      2/1/7
      VLAN                 30   e8:e7:32:00:ef:a2         dynamic     bridging      2/1/7

 Total number of Valid MAC addresses above = 4
```

Configuring MAC Address Table Aging Time

Source learning also tracks MAC address age and removes addresses from the MAC address table that have aged beyond the aging timer value. When a device stops sending packets, source learning keeps track of how much time has passed since the last packet was received on the switch port of the device. When this amount of time exceeds the aging time value, the MAC is aged out of the MAC address table. Source learning always starts tracking MAC address age from the time since the last packet was received.

By default the MAC address aging time is set to 300 seconds. This can be viewed:

```
6860-> show mac-learning aging-time
Mac Address Aging Time (seconds) = 300
```

This can be changed using the command:

```
6860-> mac-learning aging-time 500
6860-> show mac-learning aging-time
Mac Address Aging Time (seconds) = 500
```

5.2. **Advanced Troubleshooting Scenarios**

MAC Addresses Not Aging Out

There are times when idle MAC addresses are not aging out of the MAC address table and it causes the MAC address table size to increase.

Two common scenarios which cause the idle MAC addresses not to age out:
a). An interface on the switch is flapping (between every 1 to 4 minutes).
b). TCNs are received by the switch (between every 1 to 4 minutes) on one VLAN.

a). When port link status is toggled, due to spanning tree and other L2 protocol requirements, the switch needs to flush the MAC addresses associated to that port in all VLANs. During this process to synchronize HW and SW tables, the switch will block HW MAC table updates not being generated to SW. For this purpose, the port level aging feature of the Broadcom ASIC is used.
When this port level aging register is modified (when port flushes are triggered), the global aging timer is restarted.
Since the global aging timer is restarted, the MAC addresses on the other interfaces are not flushed causing the MAC address table size to increase.

b). The same global aging timer behavior applies when TCNs are receive on a VLAN especially in per VLAN STP mode.
When TCN is received on VLAN 10, the switch flushes all the MAC addresses in VLAN 10 and then reset the global aging timer causing the idle MAC addresses of all other VLANs not to be flushed.
If there is a high-rate of TCNs on VLAN 10 to the switch (e.g. >once every 1 to 4 minutes) this will cause the global aging timer (300 sec default) to keep resetting and idle MAC addresses of the other VLANs will not be aged out from the switch.

Use the below steps to troubleshoot this issue on OS6860/OS6900/OS10K.

Scenario 1:

a). An interface in the switch is flapping (at least 1 time every 1 to 4 minutes).
Use the below “show interfaces” command to find out which port is showing the continous link flaps. Check the below highlighted fields to understand which interface is recently flapping.

```
6860-> show interfaces 1/1/2
Chassis/Slot/Port  1/1/2  :
 Operational Status     : up,
 Last Time Link Changed : Sat Feb  8 00:47:56 2014,
 Number of Status Change: 9,
 Type                   : Ethernet,
 SFP/XFP                : N/A,
 EPP                    : Disabled,
 Link-Quality           : N/A,
 MAC address            : e8:e7:32:ab:1c:5f,
 BandWidth (Megabits)   :     1000,            Duplex           : Full,
 Autonegotiation        :   1  [ 1000-F 100-F 100-H 10-F 10-H ],
 Long Frame Size(Bytes) : 9216,
 Rx              :
 Bytes Received  :           785209334, Unicast Frames :              255953,
 Broadcast Frames:             8787093, M-cast Frames  :             1420762,
 UnderSize Frames:                   0, OverSize Frames:                   0,
 Lost Frames     :                   0, Error Frames   :                   0,
 CRC Error Frames:                   0, Alignments Err :                   0,
 Tx                 :
 Bytes Xmitted      :           614365318, Unicast Frames :              5726870,
 Broadcast Frames:                  45986, M-cast Frames  :               425461,
 UnderSize Frames:                      0, OverSize Frames:                    0,
 Lost Frames        :                   0, Collided Frames:                    0,
 Error Frames       :                   0
```

The below grep options is available on 7X/8X to find out which ports are going up and down at present. The output below shows that the port 1/1/4 has flapped 3655 times and port 1/1/10 has flapped 2656 times.

```
-> show interfaces | grep Number
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 3655,
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 2656,
 Number of Status Change: 1,
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 0,
 Number of Status Change: 0,
```

To determine if ports are actively flapping, start by using the “show system” command to check the current time and then use the below command to see which ports are currently flapping.

The output below shows that the port 1/1/4 and 1/1/10 has flapped recently.

```
-> show interfaces | grep Last
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  13 13:20:20 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  13 13:20:14 2015,
 Last Time Link Changed : Wed Jan  1 02:58:05 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
 Last Time Link Changed : Wed Jan  1 00:54:14 2015,
```

Scenario 2:

b). TCNs are received by the switch (at least 1 every 1 to 4 minutes) for one or more VLANs. In order to track down which VLAN and port the TCNs are received on (causing the global MAC aging timer to restart).

Use the below command to find out which VLAN is receiving the TCNs

```
6860-> show spantree vlan 200
Spanning Tree Parameters for Vlan 200
  Spanning Tree Status :                   ON,
  Protocol             :       IEEE Rapid STP,
  mode                 : Per VLAN (1 STP per Vlan),
  Priority             :       32768 (0x8000),
  Bridge ID            :   8000-e8:e7:32:ab:1c:57,
  Designated Root      :   8000-e8:e7:32:38:d0:c0,
  Cost to Root Bridge  :                    4,
  Root Port            :                1/1/2,
  Next Best Root Cost  :                    0,
  Next Best Root Port  :                 None,
  TxHoldCount          :                    3,
  Topology Changes     :                    7,
  Topology age         :             03:23:38,
    Current Parameters (seconds)
      Max Age              =    20,
      Forward Delay        =    15,
      Hello Time           =     2
    Parameters system uses when attempting to become root
      System Max Age       =    20,
      System Forward Delay =    15,
      System Hello Time    =     2
```

To find out which port is receiving the TCNs, you can use below command.

```
-->su
SHASTA#-> pidof stpNi
2076
SHASTA#-> debug 2076 "call stpni_printStats(1,1)"
[Thread debugging using libthread_db enabled]
0x0f923db0 in ___newselect_nocancel () from /lib/tls/libc.so.6
--------------------------------------------------------------------------------------------
    |               RX                   |               TX                    |    AGGR BPDU
PORT| Bpdu RBpdu MBpdu Flg80 Flg01   TCN | Bpdu RBpdu MBpdu Flg80 Flg01   TCN |   Rx     Tx
--------------------------------------------------------------------------------------------
x01:     0     2     0     0     2      0     0    30     0     0     2    0      0      0
x03:     0     1     0     0     0      0     0    27     0     0     2    0      0      0
x12:     0     2     0     0     2      0     0    30     0     0     2    0      0      0
--------------------------------------------------------------------------------------------
$1 = 1
```

Refer the “Troubleshooting STP” chapter for detailed information on troubleshooting issues related to STP.

MAC Address Flapping

There are scenarios in which one or more MAC addresses flapping between two interfaces. MAC address flapping is mostly caused by a layer 2 loop in the network (which are not detected by STP).
The command “**show mac-learning mac-address <mac>**” show if the MAC address is flapping between two ports. Refer the bshell troubleshooting section for detailed information.

```
-> show mac-learning mac-address 00:13:72:19:5e:1f
Legend: Mac Address: * = address not valid,

        Mac Address: & = duplicate static address,

   Domain     Vlan/SrvcId/ISId        Mac Address         Type          Operation     Interface
-------------+--------------------+-------------------+----------------+-------------+----------
       VLAN                  10   00:13:72:19:5e:1f         dynamic     bridging       2/1/15


-> show mac-learning mac-address 00:13:72:19:5e:1f
```

```
Legend: Mac Address: * = address not valid,

        Mac Address: & = duplicate static address,

   Domain      Vlan/SrvcId/ISId        Mac Address        Type          Operation    Interface
-----------+---------------------+-------------------+---------------+------------+----------
       VLAN                  10    00:13:72:19:5e:1f         dynamic     bridging      2/1/16
```

5.3. bShell Troubleshooting

Warning: Maintenance Shell commands should only be used by Alcatel-Lucent personnel or under the direction of Alcatel-Lucent. Misuse or failure to follow procedures that use Maintenance Shell commands in this guide correctly can cause lengthy network down time and/or permanent damage to hardware.

If a problem is specific to a NI and the MAC address is not being learned by the switch, the first step is to verify from the the NI that the MAC address has been learned. There is a possibility that the NI has learned the MAC but CMM is not reporting that MAC because of lost IPC messages between the CMM and NI.

The “l2 show” command will dump all of the MAC addresses. Check the “show mac-learning summary” command to confirm how many mac-addresses are present befoare using the “l2 show” command.

```
6860-> su
Entering maintenance shell. Type 'exit' when you are done.
SHASTA #-> bshell

Entering character mode
Escape character is '^]'.

BCM.0> l2 show
mac=00:0a:1e:22:02:51 vlan=4094 modid=4 port=0   Hit
mac=00:13:72:19:5e:1f vlan=10 modid=4 port=16   Hit
mac=e8:e7:32:00:ef:a2 vlan=10 modid=4 port=7   Hit
mac=e8:e7:32:b3:49:26 vlan=1 modid=4 port=16   Hit
mac=e8:e7:32:b3:49:27 vlan=1 modid=4 port=15   Hit
mac=00:0a:1e:22:01:f8 vlan=4094 modid=0 port=0/cpu0 Hit CPU
mac=01:20:da:99:99:99 vlan=1 modid=0 port=0/cpu0 Static CPU MCast=1
mac=e8:e7:32:00:ef:a2 vlan=1 modid=4 port=7   Hit
mac=00:0a:1e:22:02:f8 vlan=4094 modid=4 port=0   Hit
mac=e8:e7:32:00:ef:a2 vlan=30 modid=4 port=7   Hit
```

Mac-Address Lookup in bShell:-

OS6900 / OS10K bShell command to search for a MAC address in the hardware.

```
6900-> su
Entering maintenance shell. Type 'exit' when you are done.
TOR #-> bshell

Entering character mode
Escape character is '^]'

BCM.0>sear l2_entry mac_addr=0x442b032ead41
Searching L2_ENTRY table indexes 0x0 through 0x7fff...
L2_ENTRY.ipipe0[4328]:
<VPG_TYPE_1=0,VPG_TYPE=0,VPG_1=0x215,VPG=0xb,VLAN_ID=0xfc,VFI=0xfc,VALID=1,T_1=1,TGID_1=0x15,TGID
=0xb,T=1,STATIC_BIT=1,SRC_DISCARD=0,SCP=0,RPE=0,REMOTE_TRUNK_1=0,REMOTE_TRUNK=0,REMOTE=0,PRI=0,PO
RT_NUM_1=0x15,PORT_NUM=0xb,PENDING=0,OVID=0xfc,MODULE_ID_1=8,MODULE_ID=0,MIRROR=0,MAC_BLOCK_INDEX
=0,MAC_ADDR=0x442b032ead41,LOCAL_SA=0,LIMIT_COUNTED=0,L3=0,L2MC_PTR=0xb,L2:VPG_TYPE=0,L2:VPG=0xb,
L2:VLAN_ID=0xfc,L2:VFI=0xfc,L2:TGID=0xb,L2:T=1,L2:STATIC_BIT=1,L2:SRC_DISCARD=0,L2:SCP=0,L2:RPE=0
,L2:REMOTE_TRUNK=0,L2:REMOTE=0,L2:PRI=0,L2:PORT_NUM=0xb,L2:PENDING=0,L2:MODULE_ID=0,L2:MIRROR=0,L
2:MAC_BLOCK_INDEX=0,L2:MAC_ADDR=0x442b032ead41,L2:LIMIT_COUNTED=0,L2:L3=0,L2:L2MC_PTR=0xb,L2:EH_T
```

```
M=0,L2:EH_TAG_TYPE=0,L2:EH_QUEUE_TAG=0,L2:DUMMY_INDEX=0,L2:DST_DISCARD=0,L2:DEST_TYPE=1,L2:DESTIN
ATION=0xb,L2:CPU=0,L2:CLASS_ID=0,L2:ASSOCIATED_DATA=0x10000000000200b,KEY_TYPE=0,IVID=0xd41,HITSA
=0,HITDA=0,EVEN_PARITY=1,EH_TM=0,EH_TAG_TYPE=0,EH_QUEUE_TAG=0,DUMMY_INDEX=0,DST_DISCARD=0,DEST_TY
PE=1,DESTINATION_1=0x215,DESTINATION=0xb,CPU=0,CLASS_ID=0,ASSOCIATED_DATA=0x10000000000200b>
```

OS6860 bShell command to search for a MAC address in the hardware.

```
OS6860-> su
Entering maintenance shell. Type 'exit' when you are done.
SHASTA #-> bShell

Entering character mode
Escape character is '^]'.

BCM.0> search l2_entry_1 mac_addr=0x001372195e1f
Searching L2_ENTRY_1 table indexes 0x0 through 0xbfff...
L2_ENTRY_1.ism0[1276]:
<WIDE_ENTRY_BITS=0x000001802100013721 95e1f00283,WIDE=0,VLAN_ID=0xa,VIF:L2MC_PTR=0x10ca,VFI_RESERV
ED=0,VFI=0xa,VALID=1,TRILL_NETWORK_RECEIVERS_PRESENT=0,TGID=0x210,STATIC_BIT=0,SCP=0,RPE=0,RESERV
ED_102_102=0,REMOTE=1,PRI=0,PORT_NUM=0x10,PE_VID:L2MC_PTR=0x195,PENDING=0,MODULE_ID=4,MAC_BLOCK_I
NDEX=0,MAC_ADDR=0x001372195e1f,LOCAL_SA=0,LIMIT_COUNTED=1,L2MC_PTR=0x210,L2:VLAN_ID=0xa,L2:VFI_RE
SERVED=0,L2:VFI=0xa,L2:TRILL_NETWORK_RECEIVERS_PRESENT=0,L2:TGID=0x210,L2:SCP=0,L2:RPE=0,L2:RESER
VED_102_102=0,L2:REMOTE=1,L2:PRI=0,L2:PORT_NUM=0x10,L2:PENDING=0,L2:MODULE_ID=4,L2:MAC_BLOCK_INDE
X=0,L2:MAC_ADDR=0x001372195e1f,L2:LIMIT_COUNTED=1,L2:L2MC_PTR=0x210,L2:KEY=0x00004dc865787c00a0,L
2:HASH_LSB=0x5e1f,L2:DUMMY_2=0,L2:DUMMY_1=0,L2:DUMMY=0,L2:DEST_TYPE=0,L2:DESTINATION=0x210,L2:DAT
A_B=0,L2:DATA_A=0x0000180210,L2:DATA=0x0000180210,L2:CLASS_ID_MSB=0,L2:CLASS_ID_FULL=0,L2:CLASS_I
D=0,KEY_TYPE=0,KEY=0x00004dc865787c00a0,HIT_BITS=2,HITSA=1,HITDA=0,HASH_LSB=0x5e1f,EVEN_PARITY=1,
DUMMY_2=0,DUMMY_1=0,DUMMY=0,DEST_TYPE=0,DESTINATION=0x210,DATA_B=0,DATA_A=0x0000180210,DATA=0x000
0180210,CLASS_ID_MSB=0,CLASS_ID_FULL=0,CLASS_ID=0,>
```

MAC Address Continously Flushing :

Issues have been seen in our customer environments where the MAC address table is continuously flushing because of a layer-2/bridging loop. This causes the hardware to write the MAC address as static in the hardware.

For dynamic MAC addresses, the STATIC-BIT should always be zero. During a loop condition, the hardware sets the STATIC-BIT to 1 and sometimes, even after the loop has been recovered, the hardware does not update the table (i.e STATIC-BIT does not set back to zero).

This may cause the MAC address to be learned on an incorrect port in the hardware which will cause the traffic to be dropped for that destination.

```
6900-> su
Entering maintenance shell. Type 'exit' when you are done.
TOR #-> bshell

Entering character mode
Escape character is '^]'

BCM.0> search l2_entry mac_addr=0xe8e7322d2c47
Searching L2_ENTRY table indexes 0x0 through 0x1ffff...
L2_ENTRY.ipipe0[5408]:
<VPG_TYPE_1=1,VPG_TYPE=0,VPG_1=0x1ce,VPG=0x1d,VLAN_ID=1,VIF:L2MC_PTR=0x1d,VFI=1,VALID=1,T_1=1,TRI
LL_NETWORK_RECEIVERS_PRESENT=0,TGID_1=0x1ce,TGID=0x1d,T=0,STATIC_BIT=1,SRC_DISCARD=0,SCP=0,RPE=0,
RESERVED_1=0x322d2,REMOTE_TRUNK_1=1,REMOTE_TRUNK=0,PRI=0,PORT_NUM_1=0x4e,PORT_NUM=0x1d,PENDING=0,
OVID=1,MODULE_ID_1=0xa3,MODULE_ID=0,MAC_BLOCK_INDEX=0,MAC_ADDR=0xe8e7322d2c47,LOCAL_SA=1,L2MC_PTR
=0x1d,L2:VPG_TYPE=0,L2:VPG=0x1d,L2:VLAN_ID=1,L2:VFI=1,L2:TRILL_NETWORK_RECEIVERS_PRESENT=0,L2:TGI
D=0x1d,L2:T=0,L2:STATIC_BIT=1,L2:SRC_DISCARD=0,L2:SCP=0,L2:RPE=0,L2:REMOTE_TRUNK=0,L2:PRI=0,L2:PO
RT_NUM=0x1d,L2:PENDING=0,L2:MODULE_ID=0,L2:MAC_BLOCK_INDEX=0,L2:MAC_ADDR=0xe8e7322d2c47,L2:L2MC_P
TR=0x1d,L2:KEY=0x7473991696238008,L2:DUMMY_INDEX=0,L2:DST_DISCARD=0,L2:DEST_TYPE=0,L2:DESTINATION
=0x1d,L2:DATA=0x82000001d,L2:CPU=0,L2:CLASS_ID=0,L2:ASSOCIATED_DATA=0x82000001d,KEY_TYPE=0,KEY=0x
7473991696238008,IVID=0xc47,HITSA=1,HITDA=0,EVEN_PARITY=1,DUMMY_INDEX=0,DST_DISCARD=0,DEST_TYPE_1
```

```
=3,DEST_TYPE=0,DESTINATION_1=0x51ce,DESTINATION=0x1d,DATA=0x82000001d,CPU=0,CLASS_ID=0,ASSOCIATED
_DATA=0x82000001d>
```

To recover the customer network from the issue state, change the static_bit value back to zero.

```
BCM.0> mod l2_entry 5408 1 static_bit=0
```

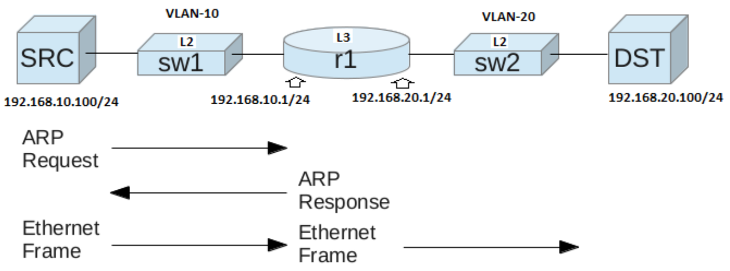
```
BCM.0> search l2_entry mac_addr=0xe8e7322d2c47
Searching L2_ENTRY table indexes 0x0 through 0x1ffff...
L2_ENTRY.ipipe0[5408]:
<VPG_TYPE_1=1,VPG_TYPE=0,VPG_1=0x1ce,VPG=0x1d,VLAN_ID=1,VIF:L2MC_PTR=0x1d,VFI=1,VALID=1,T_1=1,TRI
LL_NETWORK_RECEIVERS_PRESENT=0,TGID_1=0x1ce,TGID=0x1d,T=0,STATIC_BIT=0,SRC_DISCARD=0,SCP=0,RPE=0,
RESERVED_1=0x322d2,REMOTE_TRUNK_1=1,REMOTE_TRUNK=0,PRI=0,PORT_NUM_1=0x4e,PORT_NUM=0x1d,PENDING=0,
OVID=1,MODULE_ID_1=0xa3,MODULE_ID=0,MAC_BLOCK_INDEX=0,MAC_ADDR=0xe8e7322d2c47,LOCAL_SA=1,L2MC_PTR
=0x1d,L2:VPG_TYPE=0,L2:VPG=0x1d,L2:VLAN_ID=1,L2:VFI=1,L2:TRILL_NETWORK_RECEIVERS_PRESENT=0,L2:TGI
D=0x1d,L2:T=0,L2:STATIC_BIT=0,L2:SRC_DISCARD=0,L2:SCP=0,L2:RPE=0,L2:REMOTE_TRUNK=0,L2:PRI=0,L2:PO
RT_NUM=0x1d,L2:PENDING=0,L2:MODULE_ID=0,L2:MAC_BLOCK_INDEX=0,L2:MAC_ADDR=0xe8e7322d2c47,L2:L2MC_P
TR=0x1d,L2:KEY=0x7473991696238008,L2:DUMMY_INDEX=0,L2:DST_DISCARD=0,L2:DEST_TYPE=0,L2:DESTINATION
=0x1d,L2:DATA=0x1d,L2:CPU=0,L2:CLASS_ID=0,L2:ASSOCIATED_DATA=0x1d,KEY_TYPE=0,KEY=0x74739916962380
08,IVID=0xc47,HITSA=1,HITDA=0,EVEN_PARITY=0,DUMMY_INDEX=0,DST_DISCARD=0,DEST_TYPE_1=3,DEST_TYPE=0
,DESTINATION_1=0x51ce,DESTINATION=0x1d,DATA=0x1d,CPU=0,CLASS_ID=0,ASSOCIATED_DATA=0x1d>
BCM.0>
```

6. Troubleshooting ARP

The OmniSwitch supports Address Resolution Protocol (ARP). In order to troubleshoot issues related to ARP, a basic understanding of the ARP protocol is required.

ARP is one of the major protocols in the TCP/IP stack. The purpose of ARP is to resolve an IPv4 address (32 bit logical address) to the physical address (48 bit MAC address). Applications at the application layer use IPv4 addresses at the network layer to communicate, but at the Datalink layer, the addressing is a MAC address (48 bit Physical Address).

The purpose of Address Resolution Protocol (ARP) is to derive the MAC address of a device in your local subnet, for which you have a corresponding IPv4 address. This allows you to properly frame the IP packet with a correct MAC destination in the Ethernet header.



When SRC machine 192.168.10.100 wants to reach the DST machine 192.168.20.100, the SRC machine looks at its routing table to find the next hop. On most PCs, the default gateway is used for routing. Assume, the default gateway on SRC is 192.168.10.1 (router1), the SRC will need to learn the ARP entry for its gateway. The router1 will also need to learn the ARP for the DST machine to forward the packet receieved from SRC. If SRC machine is not able to communicate with DST machine, it could be the result of an ARP resolution failure.

Summary of the commands in this chapter is listed here:

```
show mac-learning mac-address <MAC address>
show arp <IP address>
show arp  <MAC address>
show arp summary
debug ip packet start ip-address <IP address> start timeout 2
arps
arpstat
cat /proc/alv4/stats
```

6.1. Basic Troubleshooting

To troubleshoot ARP the first step is to verify the MAC address of the SRC machine and DST machine are learned on the correct port in the correct VLAN.

```
OS6860-> show mac-learning mac-address 00:13:72:19:5e:1f
Legend: Mac Address: * = address not valid,
        Mac Address: & = duplicate static address,

   Domain     Vlan/SrvcId/ISId        Mac Address        Type          Operation    Interface
-----------+---------------------+-------------------+---------------+------------+----------
       VLAN                 10   00:13:72:19:5e:1f         dynamic     bridging      1/1/15
```

This output shows that MAC address `00:13:72:19:5e:1f`, belonging to SRC, is learned on port 1/1/15 in VLAN 10.

To obtain the MAC address required for forwarding a datagram, the layer 3 switch does the following:

- First, the layer 3 switch looks in the ARP cache for an entry that lists the MAC address for the IP address. The ARP cache maps IP addresses to MAC addresses. The cache also lists the port attached to the device. A dynamic ARP entry enters the cache when the Layer 3 switch receives an ARP reply.

- If the ARP cache does not contain an entry for the destination IP address, the Layer 3 switch broadcasts an ARP request out all its IP interfaces. The ARP request contains the IP address of the destination. If the device with the IP address is reachable by the Layer 3 switch, the device sends an ARP response containing its MAC address. The response is a unicast packet addressed directly to the layer 3 switch. The layer 3 switch places the information from the ARP response into the ARP cache.

To search for a specific ARP entry, use the following command syntax:
show arp <ip-address>

```
OS6860-VCof4-> show arp 10.255.13.26
Total 22 arp entries
 Flags (P=Proxy, A=Authentication, V=VRRP, R=REMOTE, B=BFD, H=HAVLAN, I=INTF)
 IP Addr           Hardware Addr       Type       Flags   Port     Interface   Name
-----------------+-------------------+----------+-------+--------+-----------+----------
 10.255.13.26      08:00:20:a8:f0:8a   DYNAMIC             1/1/47  vlan-172
```

To search for an ARP entry associated with a MAC address use the following command syntax:
show arp *<mac-address>*
For example:

```
OS6860-VCof4-> show arp  08:00:20:a8:f0:8a
Total 20 arp entries
 Flags (P=Proxy, A=Authentication, V=VRRP, R=REMOTE, B=BFD, H=HAVLAN, I=INTF)
 IP Addr           Hardware Addr       Type       Flags   Port     Interface   Name
-----------------+-------------------+----------+-------+--------+-----------+-------------------
 10.255.13.26      08:00:20:a8:f0:8a   DYNAMIC             1/1/47  vlan-172
```

Common Error Conditions

If an ARP is not getting resolved, the following conditions may exist:

• **A** problem with the general health of the switch or NI.
• Physical link status might not be operational.
• MAC address not learned on the port.
• ARP request not reaching the switch, which may be because:
- The workstation is not sending an ARP reply.
- The workstation is not able to understand the ARP request.
- The ARP response might have been corrupted.

- Duplicate IP addresses configured on the device in the same VLAN.
• STP TCNs may lead to an ARP table flush and high ARP rate (bursts).
• The ARP table is not synchronized between CMM and NIs
• ARP packets are generated by the CPU at high rate – As an example, a device scanning the network may force an AOS switch to generate ARP traffic at high rate due to continous requests for unresolved ARPs.

6.2. **Advanced ARP Troubleshooting**

Warning: Maintenance Shell and debug commands should only be used by Alcatel-Lucent personnel or under the direction of Alcatel-Lucent. Misuse or failure to follow procedures that use Maintenance Shell commands in this guide correctly can cause lengthy network down time and/or permanent damage to hardware.

How to troubleshoot when the MAC address is learned but there is no ARP entry

a) One endpoint is not responding or ARP packets are corrupted or lost

If the MAC address is already learned on the port and the ARP is not getting resolved further troubleshooting is required on the switch to determine if the ARP requests are reaching the switch and switch is issuing ARP replies.

Troubleshooting the ARP packets requires the use of diagnostic CLI commands. Precautions must be taken when using these commands as they are likely to dump a lot of information on the screen. Appending the “timeout” argument to the command can ensure that the amount of information is manageable.

The command to use is as follows to capture the specific packets with ip address 10.255.13.26 hitting the CPU for 2 seconds:

```
OS6860-> debug ip packet start ip-address 10.255.13.26 start timeout 2

1 1 S FLD 080020:a8f08a   ->ffffff:ffffff ARP Request 10.255.13.26->10.255.13.66
1 1 R 1/1/47 e8e732:ab1c57 ->080020:a8f08a   ARP Reply 10.255.13.66->10.255.13.26
```

The above capture shows that an ARP request came in on port 1/1/47 for ip address 10.255.13.66. The ARP reply was sent by the switch to 10.255.13.26 at MAC address 08:00:20:a8:f0:8a.

This confirms that the switch is replying to the ARP requests. The ARP cache of the endpoint should also show the correct ARP entry for the switch. If not, a sniffer should be placed between the switch and the workstation to capture the packets and determine if the packets are corrupted or if either of the devices are not responding in the correct format.

b) ARP table is full
In CLI, use command show arp summary to see the total number of ARP entries.

```
-> show arp summary
   Type               Count
-----------------+--------------
 Total                  3400
 Static                 0
 Dynamic                  3400
 Authenticated            0
 Proxy                    0
 VRRP                     0
```

When the NI ARP table is full the following message is logged in switch log: "ipni_add_arp_rexmit: Rexmit List full".

Go to Maintenance Shell and Telnet to port 5008 on the affected unit

```
OS6860-> su
Entering maintenance shell. Type 'exit' when you are done.

SHASTA #-> telnet 127.2.1.1 5008
IPNI-VRF-0>

Display "rexmit" ARPs:
IPNI-VRF-0> rexmitarps
3400 entries
```

Display all ARPs..

```
IPNI-VRF-0> arps
arp_retrieved = 1

Slot 1. NI Arp Table
 memaddr   destination        MAC      vlan  port  flags  la_hold expire  asktime asked
002321d0  192.168.10.100 00e0b1:93904e   10 2/1/7      2        0  14128        0     0 0x232228
00232260  192.168.20.100 00e0b1:93904e   20 2/1/7      2        0  21508        0     0 0x2322d0
00232308  192.168.30.100 00e0b1:93904e   30 2/1/7     42        0  21839        0     0 0x232378
```

The ‘arpstat’ output provides the detailed information about events related to ARP.

```
IPNI-VRF-0> arpstat

ARP Statistics
--------------
num arps           : 1.
arp adds           : 3.
arp add fails      : 0.
arp dels           : 1.
arp del fails      : 0.
arp flushes        : 0.
arp retrieves      : 0.
arp changes        : 51.
arp chg fails      : 0.
arp refresh        : 0.
arp ref fails      : 0.
arp ref-req sent   : 42.
arp allow_snt      : 44.
arp allow_rcv      : 9.
arp input adds     : 0.
arp input chgs     : 0.
arp input refs     : 0.
arp sl learn       : 0.
arp sl dels        : 0.
arp sl chgs        : 0.
arp expires        : 1.
arp expire del     : 1.
arp expire chg     : 0.
arp send req       : 42.
arp send rep       : 0.
arp recv req       : 0.
arp recv rep       : 0.
IPNI-VRF-0>
```

The below command to see number of gratuitous ARP statistics and other errored ARP packets.

```
OS6860-> su
Entering maintenance shell. Type 'exit' when you are done.

SHASTA #-> cat /proc/alv4/stats
gratarp = 1
gratarp err = 0
rcv = 0
rcv bad ifindex = 0
rcv missing ifindex = 0
dump drop = 0
dump size = 0
```

7. Troubleshooting Spanning Tree

Summary of the commands in this chapter is listed here:

```
show spantree
show spantree vlan <vlan-id>
show spantree ports
show spantree ports active
show interfaces <slot/port>
show vlan xx members
debug stp bpdu-stats 1 start
debug stp bpdu-stats show 1
debug $(pidof stpNi) "p stpniFlushCount"
debug $(pidof stpNi) "call stpni_printStats(1,1)"
dump chg vlan <vlan-id>
stg stp 5
```

7.1. Basic troubleshooting

To troubleshoot STP issue it is necessary to have a network diagram that depicts both the physical (cables) and logical (VLANs) configurations. It also very useful to know which ports are normally in blocking/forwarding prior to any problem.

A failure of the Spanning Tree Protocol (STP) will usually cause either a bridging loop on one or more LANs or constant reconvergence of STP. Each of these scenarios will cause several problems.

• If there is a bridging loop on the LAN there can be a broadcast storm because broadcast packets will continuously loop within the network. In addition, when a unicast address is learned on a port and toggling from one port to another in a very short period, the unicast traffic will be affected.
• If STP is constantly reconverging temporary network outages can occur as ports can reach a state where they are cycling through the 30 seconds of listening and learning as defined by 802.1D. If STP is constantly reconverging the LAN can be perpetually down.

To determine the cause of an STP problem, it is useful to first verify the configuration, especially if the network having problems has recently been installed or reconfigured.

Use the **show spantree** command to verify that STP is enabled and that all devices participating in spanning tree are running the same STP protocol.

```
OS6860-> show spantree
  Spanning Tree Path Cost Mode : AUTO
 Vlan STP Status Protocol Priority
-----+----------+--------+--------------
    1      ON       RSTP   32768 (0x8000)
 4094     OFF       RSTP   32768 (0x8000)
```

Use the **show spantree** command and specify a VLAN to verify the correct mode, designated root ID, root port, and configurable timers. The timers need to be consistent across a physical link running STP. It is also useful to note the number of Topology changes and Topology age. If topology changes are incrementing quickly, the devices pariticipating in spanning tree cannot agree who is the root bridge. This can be caused by dropped BPDUs (to be discussed later), a bridge that insists it is root regardless of received BPDUs, or a physical link going in and out of service.

```
OS6860-> show spantree vlan 2
Spanning Tree Parameters for Vlan 2
  Spanning Tree Status :                    ON,
  Protocol             :       IEEE Rapid STP,
  mode                 : Per VLAN (1 STP per Vlan),
  Priority             :       32768 (0x8000),
  Bridge ID            :   8000-e8:e7:32:b3:49:11,
  Designated Root      :   8000-00:e0:b1:93:90:4e,
  Cost to Root Bridge  :                     4,
  Root Port            :                 1/1/1,
  Next Best Root Cost  :                     4,
  Next Best Root Port  :                 1/1/3,
  TxHoldCount          :                     3,
  Topology Changes     :                     1,
  Topology age         :            00:15:38,
    Current Parameters (seconds)
      Max Age              =    20,
      Forward Delay        =    15,
      Hello Time           =     2
    Parameters system uses when attempting to become root
      System Max Age       =    20,
      System Forward Delay =    15,
      System Hello Time    =     2
```

Use the **show spantree ports** command to determine if the port is in forwarding or blocking and are in the correct VLAN. Remember that in any LAN with physical path redundancy there must be at least one port in blocking status. Knowledge of the ports which are usually in a blocking state, can be leveraged as a starting point for troubleshooting. Has their state changed?

```
OS6860-> show spantree ports
 Vlan  Port   Oper Status  Path Cost  Role    Note
-----+-------+------------+---------+-------+---------
    2   1/1/1     FORW             4   ROOT
    2   1/1/2      DIS             0    DIS
    2   1/1/3      BLK             4    ALT

OS6860-> show spantree ports active
 Vlan  Port   Oper Status  Path Cost  Role    Note
-----+-------+------------+---------+-------+---------
    1   1/1/1     FORW             4   ROOT
    1   1/1/3      BLK             4    ALT
```

If ports that should be in a blocking state are now forwarding, there are two likely causes. The first is that there was a physical failure in a link that was previously forwarding. The second is that the BPDUs from the root are being dropped. If it appears that BPDUs are being dropped, troubleshoot this as if it were any other packet being dropped.

Use the **show interfaces** command to look for errors incrementing on the port as well as to verify duplex settings match on either side of the link.

```
OS6860-> show interfaces 1/1/1
Chassis/Slot/Port  1/1/1  :
 Operational Status     : up,
 Last Time Link Changed : Wed Jan  1 00:20:13 2014,
 Number of Status Change: 1,
 Type                   : Ethernet,
 SFP/XFP                : N/A,
 EPP                    : Disabled,
 Link-Quality           : N/A,
 MAC address            : e8:e7:32:b3:49:18,
 BandWidth (Megabits)   :     1000,             Duplex           : Full,
 Autonegotiation        :   1  [ 1000-F 100-F 100-H 10-F 10-H ],
 Long Frame Size(Bytes) : 9216,
 Rx              :
```

```
Bytes Received  :                89352, Unicast Frames :                 20,
Broadcast Frames:                    0, M-cast Frames  :               1368,
UnderSize Frames:                    0, OverSize Frames:                  0,
Lost Frames     :                    0, Error Frames   :                  0,
CRC Error Frames:                    0, Alignments Err :                  0,
Tx              :
Bytes Xmitted   :                20064, Unicast Frames :                  0,
Broadcast Frames:                    0, M-cast Frames  :                311,
UnderSize Frames:                    0, OverSize Frames:                  0,
Lost Frames     :                    0, Collided Frames:                  0,
Error Frames    :                    0
```

If the problem is determined to be layer 2 data loop, it is recommended to disable all redundant links either administratively or by disconnecting cables.

7.2. **Advanced Troubleshooting**

Useful Checklist for spanning tree problems

- Make sure that all devices are running the same STP mode (1x1, FLAT, MSTP).
- If Auto Fabric is enabled then spantree mode is forced to flat.
- If using MSTP check if all devices are using the same domain name.
- If using MSTP all VLANs within an MSTI must be tagged on all interswitch links otherwise MSTP becomes unpredictable.
- If using MSTP all switches participating in the same region must have an identical MSTP configuration.
- Check latency and connectivity loss is from layer 2 or layer 3 using ping.
- Use show mac-address-table count to verify flushing and re-learning of MAC addresses on switch.
- Check whether spanning tree in flat or 1x1 mode using "show spantree".
- Determine root-bridge and make sure that it's on the right bridge.
- Use stpni_printStats to verify on which ports TCNs and Flag01 counters are incrementing on each NI (stpni_printStats may be cleared).
- Restrict TCNs on ports where Rx counters for TCN and/or Flag01 are incrementing.
- PVST+ BPDUs are affected (dropped in MC-LAG and qos user-port) only in case AOS is explicitely configured in 1x1 PVST+ mode.

Disputed State

A port in STP will be in "Disputed" state, when a port which is receiving an inferior STP BPDU (low priority) even in learning state. This means that even after a switch sends a higher priority BPDU, if it then continues to receive a lower priority STP BPDU that port will move to "Disputed" state and will be in a "Listening" state. In this state, no traffic is allowed through this port and "show vlan port" CLI, which will be in blocking state. This will not make the port link go down. Only the VLAN will be in blocking state.

To Enable detailed logs in SWLOG in case of unexpected STP state

swlog output flash-file-size 12500
swlog appid portMgrCmm subapp all level debug2
swlog appid intfCmm subapp all level debug2
swlog appid VlanMgrCmm subapp all level debug2
swlog appid portMgrNi subapp all level debug2
swlog appid VlanMgrNi subapp all level debug2

Toggle the port state to recreate the issue. Reduce the logging level to ‘info’ and gather SWLOG outputs:

```
swlog appid portMgrCmm subapp all level info
swlog appid intfCmm subapp all level info
swlog appid VlanMgrCmm subapp all level info
swlog appid portMgrNi subapp all level info
swlog appid VlanMgrNi subapp all level info
show log swlog
```

Troubleshooting by Debug Commands

BPDU statistics collection:

```
OS6860-> debug stp bpdu-stats 1 start
BPDU Statistics collection started For inst 1

OS6860-> debug stp bpdu-stats show 1
 Port   rxCfg rxRstp rxMstp rxTcn | txCfg txRstp txMstp txTcn
-------+-----+------+------+------+------+------+------+-----
  1/1/1     0      0      0      0      0      0      0      0
  1/1/2     0      0      0      0      0      0      0      0
  1/1/3     0     11      0      0      0   1146      0      0
  1/1/4     0      0      0      0      0      0      0      0
  1/1/5     0      0      0      0      0      0      0      0
  1/1/6     0      0      0      0      0      0      0      0
  1/1/7     0     12      0      0      0   1133      0      0
  1/1/8     0      0      0      0      0      0      0      0
  1/1/9     0      5      0      0      0   1156      0      0
 1/1/10     0      0      0      0      0      0      0      0

OS6860-> debug stp bpdu-stats 1 stop
BPDU Statistics collection  Stopped
```

Troubleshooting in Maintenance Shell Commands

Warning: Maintenance Shell commands should only be used by Alcatel-Lucent personnel or under the direction of Alcatel-Lucent. Misuse or failure to follow procedures that use Maintenance Shell commands in this guide correctly can cause lengthy network down time and/or permanent damage to hardware.

To dispay the number of flush events:

```
SHASTA #-> pidof stpNi
2100
SHASTA #-> debug 2100 "p stpniFlushCount"
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/libthread_db.so.1".
0xb6a8ecbc in epoll_wait () from /lib/libc.so.6
$1 = 42
```

Command stpni_printStats :

The usual command (show spantree [VlanId]) only shows the real topology changes in the network. The aim of the commands introduced below is to view the topology changes frames (TCN) received by the switch that are not causing a change in the Spanning Tree (STP) Topology Counter displayed with show spantree [VlanId].

The commands can be used to find the sources of Topology Change STP frames (BPDUs) in a network. These topology changes make the switch clear its MAC address table according to the spanning tree protocol, and consequently its ARP table. This can increase the CPU load on the switch. The source of the Topology Change can be a switch added by the users of the network, a Blade Centre that includes internal switches, etc. This can also be a configuration error on switches cause make them send BPDUs with topology change notifications when a port dedicated to users changes status. In that case each time a user powers up or disconnects his computer, a topology change will be sent.

For STP mode 1x1 please use (vid,1) as the argument

```
SHASTA #-> debug $(pidof stpNi) "call stpni_printStats(1,1)"
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/libthread_db.so.1".
0xb6a31cbc in epoll_wait () from /lib/libc.so.6

--------------------------------------------------------------------------------------------
    |               RX                   |                TX                   |    AGGR BPDU
PORT| Bpdu RBpdu MBpdu Flg80 Flg01   TCN | Bpdu RBpdu MBpdu Flg80 Flg01   TCN |   Rx     Tx
--------------------------------------------------------------------------------------------
x00:     0   124     0     0     4     0      0     9     0     0     4      0      0      0
x02:     0   124     0     0     4     0      0     5     0     0     2      0      0      0
--------------------------------------------------------------------------------------------
$1 = 1
```

For STP and MSTP mode flat please use (0,0) as the argument

```
SHASTA #-> debug $(pidof stpNi) "call stpni_printStats(0,0)"
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/libthread_db.so.1".
0xb6a31cbc in epoll_wait () from /lib/libc.so.6

--------------------------------------------------------------------------------------------
    |               RX                   |                TX                   |    AGGR BPDU
PORT| Bpdu RBpdu MBpdu Flg80 Flg01   TCN | Bpdu RBpdu MBpdu Flg80 Flg01   TCN |   Rx     Tx
--------------------------------------------------------------------------------------------
x00:     0   289     0     0     6     0      0    11     0     0     6      0      0      0
x02:     0   289     0     0     6     0      0     7     0     0     3      0      0      0
--------------------------------------------------------------------------------------------
$1 = 1
```

For MSTI instance 1 please use (1,0) as the argument. { (instance,0) 0 represents the flat mode }

```
SHASTA #-> debug $(pidof stpNi) "call stpni_printStats(1,0)"
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/libthread_db.so.1".
0xb6a31cbc in epoll_wait () from /lib/libc.so.6

--------------------------------------------------------------------------------------------
    |               RX                   |                TX                   |    AGGR BPDU
PORT| Bpdu RBpdu MBpdu Flg80 Flg01   TCN | Bpdu RBpdu MBpdu Flg80 Flg01   TCN |   Rx     Tx
--------------------------------------------------------------------------------------------
x00:     0   289     0     0     6     0      0    11     0     0     6      0      0      0
x02:     0   289     0     0     6     0      0     7     0     0     3      0      0      0
--------------------------------------------------------------------------------------------
$1 = 1
```

Column description:
RX - All the BPDU received by the swith

- Bpdu - 802.1d BPDUs
- RBpdu - 802.1w BPDUs
- Flg80 - BPDUs with flag 80 set (Topology Change Acknowledgement)
- Flg01 - BPDUs with flag 01 set (Topology Change)

TX: All the BPDU sent by the swith

- Bpdu - 802.1d BPDUs
- RBpd - 802.1w BPDUs
- Flg80 - BPDUs with flag 80 set (Topology Change Acknowledgement)
- Flg01 - BPDUs with flag 01 set (Topology Change)

Troubleshooting in bShell

Verify a VLAN's Spanning Tree Group (STG):

```
BCM.0> dump chg vlan 2
VLAN.ipipe0[2]:
<VLAN_PROFILE_PTR=1,VALID=1,STG=5,PORT_BITMAP_W1=0xc00000,PORT_BITMAP_W0=0x5f,PORT_BITMAP=0x00c00
0000000005f,ING_PORT_BITMAP_W1=0xc00000,ING_PORT_BITMAP_W0=0x5f,ING_PORT_BITMAP=0x00c00000000000005
f,EVEN_PARITY_1=1,EVEN_PARITY_0=1,ENABLE_IGMP_MLD_SNOOPING=1,>
BCM.0>
```

Convert the port bitmat to readable format:

```
BCM.0> pbmp 0x5f
    0x000000000000000000000000000000000000000000000000000000000000005f ==>
cpu,ge0-ge4
BCM.0>
```

```
    OS6860-> show vlan 2 members
       port      type        status
    ----------+-----------+---------------
      1/1/1      default     forwarding
      1/1/2      default       inactive
      1/1/3      default       blocking
      1/1/4      default       inactive
      1/1/5      default       inactive
```

Verify the STG state in hardware to check if the port is mapped and configured correctly:

```
    BCM.0> stg stp 5
    STG 5:
        Block: ge1-ge28,xe
      Forward: ge0,hg
    BCM.0>
```

```
    OS6860-> show spantree ports blocking
     Vlan  Port   Oper Status  Path Cost  Role    Note
    -----+-------+------------+---------+-------+---------
        2   1/1/3      BLK             4    ALT

    OS6860-> show spantree ports forwarding
     Vlan  Port   Oper Status  Path Cost  Role    Note
    -----+-------+------------+---------+-------+---------
        2   1/1/1     FORW             4   ROOT
```

8. Troubleshooting Link Aggregation

Special frames (Link Aggregation Control Protocol Data Unit or LACPDU) are used to interact with a remote system and establish a Link Aggregation Group. However, some preliminary configuration is still needed, whereby the user specifies attributes related to the aggregate group as well as attributes associated to the "possible" participating links. These attributes are normally named "keys". Links will join aggregate groups by the time they come up and exchange LACPDU frames with the peer. Depending on the "keys" associated to the links and to the aggregate, the links may potentially join different groups. Note that a link can only join an aggregate when it is up because LACPDU must be exchanged with a remote system.

- For load balancing purposes, the traffic is distributed across all the ports of an aggregate group.
- The load balancing is performed at the ingress side
- The speed of the ports is not taken into consideration when traffic is distributed. In other words, the same number of flows is distributed evenly on each port without reference to the line speed.
- The maximum number of aggregates per system is 128 and the maximum number of aggregable ports is 256, including LACP aggregate.
- The maximum number of links per aggregate is 8, but an aggregate may be defined with a maximum of 8, 4 or 2 links). If no aggregate size is provided at configuration time, the default value is 8.
- From the perspective of the LACP state machine, each port can have the following states:
  - Configured: the port has been created and keeps trying to select an aggregate on the NI where the port is located
  - Selected: the port is selected in an aggregate based on the match of actor/partner parameters for the port and the aggregate, each aggregate can have up to 8 ports selected
  - Reserved: the necessary resources to handle the port are available and the port is waiting for LACPDU synchronization exchange to open the traffic

- In case of linkagg port leave / join events verify LACP BPDU drop on software and hardware level, finally compare number of LACP BPDUs received in hardware and software
- Unknown Destination/Broadcast/Multicast traffic is load balanced on all the ports of the aggregate. This provides better throughput for broadcast and multicast traffic.

- LACP BPDUs are tunneled by default on UNI ports.

```
OS6860-> show ethernet-service uni-profile default-uni-profile
   Profile Name       Stp     802.1x   802.3ad   802.1ab    MVRP     AMAP
-------------------+--------+--------+---------+---------+--------+---------
default-uni-profile  tunnel   tunnel   tunnel    tunnel    tunnel   tunnel
```

An example configuration with allowed peering:

```
OS6860-> show configuration snapshot linkagg
! Link Aggregate:
linkagg lacp agg 1 size 2 admin-state enable
linkagg lacp agg 1 actor admin-key 1
linkagg lacp port 1/1/1 actor admin-key 1
linkagg lacp port 1/1/2 actor admin-key 1
```

LACP BPDU processing on software level

Once the packets were classified as "trap to CPU" on the ASIC level the packets were placed in CPU port queue. There are 32 internal queues out of which 27 are used for LACP. The pktDriver receives the packet via DMA transfer from the Broadcom ASIC, classifies the packet as LACP, and sends the packet to the LACP module for further software processing.

Hash-control

Hash may be controlled for each linkagg separately using the following command:

```
OS6860-> linkagg lacp agg 1 hash ?
                          ^
                          TUNNEL-PROTOCOL SOURCE-AND-DESTINATION-MAC
                          SOURCE-AND-DESTINATION-IP SOURCE-MAC SOURCE-IP
                          DESTINATION-MAC DESTINATION-IP
 (Link Aggregation Command Set)
```

Support of 256 aggregates and up to 16 ports

A debug CLI command needs to be enabled to support 256 aggregates and up to 16 ports:

```
OS6860-> debug capability linkagg increase-agg-limit
```

Summary of the commands in this chapter is listed here:

```
show linkagg
show linkagg agg <linkagg id>
show linkagg agg <linkagg id> port
show configuration snapshot linkagg
show log swlog | grep LACP
show log swlog | grep linkagg
debug show lacp counters port <slot/port>
debug $(pidof lacpNi) "call la_ni_lacp_port_stats_prt (-1)"
d chg trunk_group
show c ge28
g RDBGC0.ge28
g drop_pkt_cnt_ing.ge28
```

8.1. **Basic Troubleshooting**

show linkagg

```
-> show linkagg
Number Aggregate SNMP Id Size Admin State Oper State Att/Sel Ports
-------+----------+--------+----+-------------+-------------+--------+-----
1   Static     40000001    8 ENABLED         UP           2        2
2   Dynamic    40000002    4 ENABLED         DOWN         0        0
```

```
3   Dynamic    40000003    8 ENABLED         DOWN          0        2
4   Dynamic    40000004    8 ENABLED         UP            3        3
5   Static     40000005    2 DISABLED        DOWN          0        0
```

show linkagg <id>

```
-> show linkagg agg 1
Static Aggregate
SNMP Id : 40000001,
Aggregate Number : 1,
SNMP Descriptor : Omnichannel Aggregate Number 1 ref 40000001 size 4,
Name : ,
Admin State : ENABLED,
Operational State : UP,
Aggregate Size : 4,
Number of Selected Ports : 4,
Number of Reserved Ports : 4,
Number of Attached Ports : 4,
Primary Port : 1/1/1
```

show linkagg port

```
-> show linkagg agg 1-5 port
Slot/Port Aggregate SNMP Id Status          Agg   Oper Link Prim
---------+---------+--------+--------------+------+----+----+---------
1/1/16     Static   2016 CONFIGURED            1      UP   UP   YES
1/1/17     Static   2017 CONFIGURED          2      UP   UP    NO
3/1/1      Static   3001 CONFIGURED          3      UP   UP    NO
3/1/2      Static   3045 CONFIGURED          4      UP   UP    NO
3/1/3      Static   3069 CONFIGURED          5      UP   UP    NO
```

8.2. **Advanced Troubleshooting**

Check the LACP Rx and Tx counters :

```
OS6860-> debug show lacp counters port 1/1/1
Slot/Port   LACP Tx      LACP Rx      LACP Err    Marker RTx    Marker Rx
---------+------------+------------+------------+------------+-------------
1/1/1           126          125            0             0             0
```

Detailed LACP logs:

```
swlog appid linkAggCmm subapp all level debug1
swlog appid linkAggNi subapp all level debug1
```

Even more detailed LACP logs:

```
swlog appid linkAggCmm subapp all level debug3
swlog appid linkAggNi subapp all level debug3
```

Optionally disable duplicates and increase SWLOG size:

```
no swlog duplicate-detect
swlog output flash-file-size 512
```

Gather logs:

```
show configuration snapshot linkagg
show linkagg port
show log swlog | grep LACP
show log swlog | grep linkagg
```

The debug logs should be disabled after gathering outputs:

```
swlog appid linkAggCmm subapp all level info
swlog appid linkAggNi subapp all level info
```

Troubleshooting in Maintenance Shell

Warning: Maintenance Shell commands should only be used by Alcatel-Lucent personnel or under the direction of Alcatel-Lucent. Misuse or failure to follow procedures that use Maintenance Shell commands in this guide correctly can cause lengthy network down time and/or permanent damage to hardware.

Print LACP details per port from NI module to check if the ports are mapped correctly to the configured Agg:

```
OS6860 -> su
Entering maintenance shell. Type 'exit' when you are done.
SHASTA #-> debug $(pidof lacpNi) "call la_ni_port_prt (-1)"
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/libthread_db.so.1".
0xb6a93cbc in epoll_wait () from /lib/libc.so.6

   2/1/29 -> 0x000cad78 status=6 ifdx=101029 id=131100 type=1 agg_id=66 port_index=1 Bw:1G
             adminstate=1 operstate=2 link_up_down=1 activation_order=2 agg_ctx_p=0xca750
             mclag=0 vfl=0 req_token=0 WTR_ptr:status: (nil):514352

   Actor   : Sys ID=[e8:e7:32:b3:35:a3] Sys Prio=0 Port=131100 Port Prio=0
             Admin Key=66 Oper Key=66
             Admin State=(act1.tim1.agg1.syn0.col0.dis0.def1.exp0)
             Oper State =(act1.tim1.agg1.syn0.col0.dis0.def0.exp0)
   Partner : Sys ID=[e8:e7:32:b3:36:9b] Sys Prio=0 Key=66 Port=9245 Port Prio=0
             Admin Key=0 Oper Key=66
             Admin Sys ID=[00:00:00:00:00:00] Admin Sys Prio=0 Admin Port=0 Admin Port Prio=0
             Admin State=(act0.tim0.agg1.syn0.col1.dis1.def1.exp0)
             Oper State =(act1.tim1.agg1.syn0.col0.dis0.def1.exp1)
    Explicit SysId: 0
    Bit order : 0
```

Print LACP statistics from NI module to check if the port is receiving & transmitting LACP PDUs. There is a possibility that the packet driver dropping the packet before sending it to the LACP module. We can check the counters in the pktDrv side as well as in the LACP module side to identify if there is any difference in TX and RX packet count exist or not. Following command can be useful to identify whether there is any drop at CPU

```
SHASTA #-> debug $(pidof lacpNi) "call la_ni_lacp_port_stats_prt (-1)"
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/libthread_db.so.1".
0xb6a81cbc in epoll_wait () from /lib/libc.so.6

 1:29
       lacpdus_rx              = 106372
       marker_pdus_rx          = 0
       marker_response_pdus_rx = 0
       unknown_rx              = 0
       illegal_rx              = 0
       lacpdus_tx              = 106382
       marker_pdus_tx          = 0
       marker_response_pdus_tx = 0
       pktdrv_retry            = 0
       pktdrv_drop             = 0
       sysid_drop_tx           = 0
       sysid_drop_rx           = 0
```

In the above LCAP NI output port 1/1/29 is receiving the LACP PDUs.

Troubleshooting in BShell (the Hardware debug level)

Display linkagg information:

```
BCM.0> d chg trunk_group
TRUNK_GROUP.ipipe0[3]: <TG_SIZE=1,RTAG=6,EVEN_PARITY=1,BASE_PTR=5>
TRUNK_GROUP.ipipe0[66]: <RTAG=6,>
TRUNK_GROUP.ipipe0[67]: <RTAG=6,EVEN_PARITY=1,BASE_PTR=4>
TRUNK_GROUP.ipipe0[100]: <RTAG=6,EVEN_PARITY=1,BASE_PTR=1>
TRUNK_GROUP.ipipe0[101]: <RTAG=6,EVEN_PARITY=1,BASE_PTR=2>
```

Verifyng LACP BPDU drop on hardware level :

The following command can be used to check status before or after an issue occurs and for monitoring purposes to see is there any packets drop at the hardware level, and what type of packets are dropped.

Dump counters for the port used in the linkagg (in this case it's 1/1/29 corresponding to port ge28) 3 times in 30 second interval:

```
BCM.0> show c ge28
RUC.ge28                :                 2,119              +2,119
RDBGC0.ge28             :               119,073            +119,073               1/s
RDBGC1.ge28             :                 2,237              +2,237               2/s
RDBGC3.ge28             :                    24                 +24
ING_NIV_RX_FR.ge28      :                 5,901              +5,901               2/s
TDBGC3.ge28             :                   300                +300
R64.ge28                :                 7,210              +7,210
R127.ge28               :                 5,597              +5,597               1/s
R255.ge28               :               111,039            +111,039               2/s
...
```

Dump RDBGC0 counter 3 times in 30 second interval:

```
BCM.0> g RDBGC0.ge28
RDBGC0.ge28[1][0x38002b1e]=0x1d3cf: <EVEN_PARITY=0,COUNT=0x1d3cf>
```

Dump DROP_PKT_CNT_ING counter 3 times in 30 second interval:

```
BCM.0> g drop_pkt_cnt_ing.ge28
DROP_PKT_CNT_ING.ge28[3][0x8001001d]=0: <COUNT=0>
```

9. Troubleshooting BOOTP/DHCP/UDP Relay

Summary of the commands in this chapter is listed here:

```
show configuration snapshot ip-helper
show ip helper statistics
show ip udp relay
show ip udp relay statistics
show configuration snapshot system
show log swlog | grep -E "udpRelay|udprelay
debug ip packet protocol udp start timeout 60
cat /proc/pktdrv | grep -E "Classified 24"
```

9.1. Troubleshooting DHCP

Minimum working configuration

DHCP Relay Agent Service

```
-> show configuration snapshot ip-helper
! UDP Relay:
ip helper address 192.168.10.254
```

Generic UDP Relay Service

```
-> show configuration snapshot ip-helper
! UDP Relay:
ip helper address 192.168.10.254
ip udp relay port 53
```

DHCPv6 is a network protocol that is used for configuring IPv6 hosts with IP addresses, IP prefixes and/or other configuration required to operate on an IPv6 network.
IPv6 hosts can acquire IP addresses using stateless or stateful address autoconfiguration. DHCP tends to be preferred at sites where central management of hosts is valued; stateless autoconfiguration does not require any sort of central management, and is therefore preferable in networks where no management is readily available, such as a typical home network.

Stateless Autoconfiguration

The stateless mechanism allows a host to generate its own addresses using a combination of locally available information and information advertised by routers. The stateless approach is used when a site is not particularly concerned with the exact addresses hosts use, so long as they are unique and properly routable.
Stateless Address Autoconfiguration is used to configure both link-local addresses and additional non-link-local addresses by exchanging Router Solicitation and Router Advertisement messages with neighboring routers.
The following are the two approaches with which an IPv6 node can configure its address in a stateless fashion:

- Using automatic address configuration with prefix discovery: This is based on RFC2462. If the ‘autonomous’ flag of a Prefix Information Option contained in a router advertisement is set, the IPv6 host may automatically generate its global IPv6 address by appending its 64-bit interface identifier to the prefix contained in the router advertisement.
- Stateless DHCPv6: This is not mentioned as an option given in router advertisements [RFC2461].

Stateful Autoconfiguration

In the stateful address auto-configuration model, hosts obtain interface addresses and/or configuration information and parameters from a server. The stateful approach is used when a site requires tighter control over exact address assignments. Stateful Address Autoconfiguration is used to configure non-link-local addresses through the use of a configuration protocol such as DHCP.

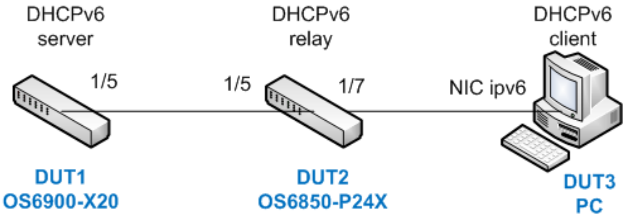
As far as the IPv6 host is concerned, using stateful DHCPv6 is little different to using stateless

DHCPv6 as the observed request/response times should be the same in most cases. However, it is possible that the extra overhead of reading and writing state to memory inside the DHCPv6 server may lead to a small increase in latency when compared to its stateless equivalent. This may be important for the configuration time of mobile nodes, which must perform address configuration when moving into a new network.

Delegating a prefix to an entire site is commonly a stateful operation, as the service provider routing scheme must always know where a site topologically resides, a packet targeted to a site must be routed back to the site. DHCPv6 server typically stores the DHCPv6 delegated prefix.

Example of Stateful Autoconfiguration

Network Diagram



Configuration Files of DHCPv6 Server

/flash/switch/ dhcpdv6.conf

```
v6-server-identifier dhcpv6;
     duid-pool {
          00-01-00-01-19-c3-8e-31-00-24-81-18-5b-f8    <---DUID of DHCPv6 client
          00-01-*
       }
       v6-subnet 2001:db7::/64 {
          policy send-unicast-option-enabled false;
        policy subnet-unavailable-threshold 90;
        policy subnet-unavailable-descent-threshold 85;
        policy minimum-requested-lifetime 800;
        option renewal-time 600;
        option rebinding-time 600;
        option preferred-lifetime 600;
        option valid-lifetime 600;
        option dns-recursive-name-server 2001:db8::1;
        v6-dynamic-dhcp range 2001:db7::2 2001:db7::ff
         {
          policy minimum-requested-lifetime 650;
          policy rapid-commit-enabled true;
          option dns-recursive-name-server 2001:db8::1;
          option domain-search-list example.com;
        }
       }
```

/flash/switch/ dhcpdv6.pcy

```
;QIPOrgID=4
AbusiveClientMonitorPeriod=30
AbusiveClientWarningCount=30
AbusiveClientLockout=30
AddManualToGlobalDuidPool=1
AllowClientPacketsWithInvalidOptions=1
AllowUnencodedFqdn=1
ClientFqdnOptionSupport=client
DefaultLease=6000
DHCPv6SocketAddr=2001:db8::1
RegisteredClientOnly=0
ReplyToUnmanagedInformationRequests=1
SendRequestedParamsOnly=1
```

```
SendUnicastOption=1
UpdateQIP=all
```

Configuration

DUT1 (OS6900-X20 running AOS 7.3.3.292.R01)

```
! VLAN:
vlan 1 admin-state enable
vlan 100 admin-state enable
vlan 100 members port 1/5 untagged
! IPv6:
ipv6 interface "dhcpv6_server" vlan 100
ipv6 address 2001:db8::1/64 "dhcpv6_server"
! IP Route Manager:
ipv6 static-route 2001:db7::/64 gateway 2001:db8::ff metric 1 dhcpv6_server
! Dhcpv6Srv:
dhcpv6-server enable
```

DUT2 (OS6850-P24X running AOS 6.4.5.447.R02)

```
! VLAN :
vlan 1 enable name "VLAN 1"
vlan 100 enable name "VLAN 100"
vlan 100 port default 1/5
vlan 200 enable name "VLAN 200"
vlan 200 port default 1/7
! VLAN SL:
! IP :
ip service all
ip interface dhcp-client vlan 1 ifindex 1
ip interface "dhcpv6_server" ifindex 2
! IPv6 :
ipv6 interface "dhcpv6_server" vlan 100
ipv6 address 2001:db8::ff/64 "dhcpv6_server"
ipv6 interface "dhcpv6_client" vlan 200 ra-managed-config-flag true
ipv6 address 2001:db7::ff/64 "dhcpv6_client"
ipv6 prefix 2001:db7::/64 autonomous-flag false dhcpv6_client
! IP multicast :
ipv6 static-route ::/0 gateway 2001:db8::ff metric 1 dhcpv6_server
! OSPF3 :
ipv6 load ospf
ipv6 ospf area 0.0.0.0
ipv6 ospf interface dhcpv6_server area 0.0.0.0
! DHCPv6 :
ipv6 helper address 2001:db8::1
```

DUT3 (PC running Windows Server 2008 R2 Entreprise)

Run "`Command Prompt`" as administrator and issue "`netsh int ipv6 show int ipv6`".
Make sure that "`Advertising = disabled`" and "`Managed Address Configuration = enabled`".
If not, issue "`netsh int ipv6 set int ipv6 adver=en managed=en`"
and then issue "`netsh int ipv6 set int ipv6 adver=dis`".

```
Interface ipv6 Parameters
----------------------------------------------
IfLuid                             : ethernet_10
IfIndex                            : 19
State                              : disconnected
Metric                             : 5
Link MTU                           : 1500 bytes
Reachable Time                     : 35000 ms
Base Reachable Time                : 30000 ms
Retransmission Interval            : 1000 ms
DAD Transmits                      : 1
Site Prefix Length                 : 64
Site Id                            : 1
Forwarding                         : disabled
Advertising                        : disabled
Neighbor Discovery                 : enabled
Neighbor Unreachability Detection  : enabled
Router Discovery                   : enabled
```

```
Managed Address Configuration      : enabled
Other Stateful Configuration       : enabled
Weak Host Sends                    : disabled
Weak Host Receives                 : disabled
Use Automatic Metric               : enabled
Ignore Default Routes              : disabled
Advertised Router Lifetime         : 1800 seconds
Advertise Default Route            : enabled
Current Hop Limit                  : 64
Force ARPND Wake up patterns       : disabled
Directed MAC Wake up patterns      : disabled
```

Output of DHCPv6 Client

```
Ethernet adapter ipv6 :

   Connection-specific DNS Suffix . . . . : example.com
   Description. . . . . . . . . . . . . . : Intel(R) PRO/1000 PT Dual Port Server Adapter #2
   Physical Address . . . . . . . . . . . : 00-15-17-D0-43-1F
   DHCP Enabled . . . . . . . . . . . . . : Yes
   Autoconfiguration Enabled. . . . . . . : Yes
   IPv6 Adress . . . . . . . . . . . . . .: 2001:db7::2(Preferred)
   Lease Obtained . . . . . . . . . . . . : Wednesday, October 30, 2013 2:20:27PM
   Lease Expires. . . . . . . . . . . . . : Wednesday, October 30, 2013 2:30:26PM
   Link-local IPv6 Address. . . . . . . . : fe80::65a9:6bca:5714:a211%14(préféré)
   IPv4 Address . . . . . . . . . . . . . : 169.254.162.17(Preferred)
   Subnet Mask. . . . . . . . . . . . . . : 255.255.0.0
   Default Gateway. . . . . . . . . . . . : fe80::eae7:32ff:fe13:647e%14
   DHCPv6 IAID. . . . . . . . . . . . . . : 251663639
   DHCPv6 Client DUID . . . . . . . . . . : 00-01-00-01-19-C3-8E-31-00-24-81-18-5B-F8
   DNS Servers . . .  . . . . . . . . . . : 2001:db8::1
   NetBIOS over Tcpip . . . . . . . . . . : Enabled
   DNS Suffix Search List. . .  . . . . . : example.com
```

9.2. **Troubleshooting UDP Relay**

Minimum working configuration

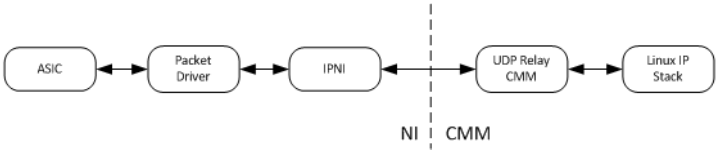
DHCP Relay Agent Service

```
-> show configuration snapshot ip-helper
! UDP Relay:
ip helper address 192.168.10.254
```

Generic UDP Relay Service

```
-> show configuration snapshot ip-helper
! UDP Relay:
ip helper address 192.168.10.254
ip udp relay port 53
```

Packet Flow



This packet flow assumes the end station is directly connected to the OS10K/OS6900/OS6860/OS6860E unit. In the customer network, the OS10K/OS6900/OS6860/OS6860E unit will most likely not directly connect to end stations but to edge switches. This packet flow description is described what happens when the UDP packets entered the OS10K/OS6900/OS6860/OS6860E unit. Also assume the end station is on VLAN 30 on VRF 2 with the ip interface for VLAN 30 is 10.30.0.254.

DHCP Relay Agent Service packet flow

1. Device sends out DHCP Discovery packet
2. DHCP Discovery packet is trapped to CPU by the FFP
3. Packet Driver sends the packet to IPNI
4. IPNI sends the packet to UDP Relay CMM based on the UDP list
5. UDP Relay CMM received this packet from the socket opened
6. UDP Relay CMM looks up the VRF that VLAN 30 belongs to
7. UDP Relay CMM looks up the next hop IP address form its configuration
8. For Standard Mode, it will be all the next hop IP addresses
9. For Per VLAN Mode, it will look up if there is any next hop IP address configured for VLAN 30; f there is none, the packet is discarded
10. Packet is sent to the Native Linux IP stack to be forwarded to the IP destination (the next hop IP address)
11. DHCP server reply with a DHCP Offer packet
12. This DHCP Offer packet is sent to ip interface 10.30.0.254
13. DHCP Offer packet is sent to the IP NI since the DHCP Server is to reply to the ip interface of the DHCP Relay Agent
14. UDP Relay CMM looks up the network port that based on the destination MAC address of the DHCP Offer packet
15. UDP Relay CMM looks up the NI that the network port resides
16. UDP Relay CMM sends the DHCP Offer packet to the IP NI where the network port is
17. IP NI sends the frame to Packet Driver where the packet is sent out into the network
18. All upstream DHCP packets from the device to the DHCP server follow the DHCP Discovery packet
19. All downstream DHCP packets from the DHCP server to the device follow the DHCP Offer packet

Generic UDP Relay Service packet flow

1. Device sends out a broadcast UDP frame with destination port 1122
2. Packet is trapped to CPU based on the destination UDP port
3. Packet is sent to IPNI from Packet Driver
4. IPNI sends the packet to UDP Relay CMM
5. UDP Relay CMM looks up the VRF of VLAN 30
6. UDP Relay CMM looks up the configuration for the UDP port 1122 and determined that this packet is to be forwarded to VLAN 40, 50 and 60
7. UDP Relay CMM will send the packet to IP NI and then to the Packet Driver 3 times (one for VLAN 40, one for VLAN 50 and one for VLAN 60)
8. Packet Driver is able to flood the packet to all the ports that belongs to VLAN 40, VLAN 50 and VLAN 60

DHCP Snooping Binding Database

By default, once DHCP Snooping is enabled at either the switch-level or the VLAN-level, the DHCP Snooping Binding Database capability will be enabled. The DHCP Snooping Binding table is indexed by the physical port and the client’s MAC address. It contains the following data:

- Client’s MAC Address;
- Client’s IP Address assigned by the DHCP Server;
- The physical port where the DHCP request is coming from;
- The VLAN Id where the DHCP request packet is coming from;
- The lease time of the IP Address;
- The type/nature of how the binding entry is populated, either static or dynamic.

The binding table entries are usually populated by the UDP Relay software as it tracks the DHCP packets against the client H/W MAC address and the physical port. It does not require any human intervention. This type is called "dynamic" (dynamically learned). When the binding entry, for any specific reason, is created by a human admin, the type is called "static" (statically configured). The dynamic binding entries take precedence over the static entries. That is, if there exists a static binding entry in the binding table, it will be replaced by a newly learned dynamic entry; while if there exists a dynamic entry, when the user attempts to add a static entry with the same MAC Address and Slot/Port, the dynamic entry is not replaced.
Since the DHCP snooping binding database needs to be persistent to survive the switch reboot/takeover, the snooping binding table is periodically saved to a file. It is named dhcpBinding.db under the /flash/switch directory. The synchronization period is configurable, and by default is 300 seconds. In addition, there will be a timestamp stating the last time the synchronization has been successfully performed. This file is also sent to the secondary CMM in a dual-CMM setup. This will have to be sent to the other chassis in a virtual chassis environment.
The dynamic binding entry is populated when the Relay Agent receives a DHCP-ACK packet. By default the Relay Agent will remove a binding entry when one of the following conditions occurs:

- Receiving a DHCP Release packet (Note, it is commonly seen that the Relay Agent does not receive the DHCP-RELEASE packets on Windows when ipconfig /release is performed)
- When the Relay Agent’s Lease Timer is decremented to 0;
- Receiving a NI-Detach event from port manager
- Receiving a link-down event from port manager
- If the MAC is aged out by source learning; This check is made at the time we sync the binding database to a file

If binding persistency is enabled by the user (default is disabled) then the only events that will cause the binding entry to be removed are receiving a DCHP-RELEASE packet or the expiration of the lease timer. The other events that normally cause removal will be ignored.

Note: Due to the synchronization period, there will potentially be a discrepancy between the binding database in the memory and the flash binding database file. Also, for the same reason the binding table in the memory might not be removed promptly, since the MAC Address aging is only checked every synchronization time period.

There are two actions defined against the DHCP Snooping binding database. The purpose of those actions is mainly for re-synchronization of the binding table (in memory) and the database (in flash).

- The "Purge" action is to clear what’s in the memory;
- The "Renew" action is to populate the binding table in the memory based on the flash file.
- Functional description:
- The max number of Binding entries in the DHCP Snooping Binding Table is 4096. (This is a soft limit that is put in place for entries syncing to the secondary and/or slave chassis).
- DHCP Snooping Binding Table on the Master primary chassis resides in memory. This table will be sync to flash based on the value of dhcpSnoopingBindingDatabasesyncTimeout value. The default is 5 minutes. The lowest value is 1 minute.
- Once DHCP Snooping Binding Table is written to flash on the Master primary CMM, the system will sync this to all the secondary/slave CMMs.
- If before the next sync to flash operation, there is a takeover action the new binding entries that are still in memory will not be saved to flash. The new Master primary CMM will not have the new entries.
- The DHCP Snooping Binding Table Persistent flag is set as disable by default same as 6.X.
- Before writing to flash, the system will decrement lease time of each entry in the DHCP Snooping Binding Table that is in memory. The system will delete those entries that the lease time expired.
- When the dhcpSnoopingBindingDatabasesyncTimeout is changed, the previous timer is stopped and the system will execute the timeout out with respect to the time that the timeout value is changed. (Start from fresh).
- Ingress Source Filtering can only be enabled on the "client-only" ports.

Configuration

DHCP Relay Agent Service

```
ip helper <standard | per-vlan>
ip helper address 172.6.5.1 vlan 1
no ip helper address 172.6.5.1                              ! standard mode
no ip helper address vlan 20 address 172.6.20.1             ! per VLAN mode.
ip helper forward delay
ip helper maximum hops
ip helper address 210.10.1.100 162.3.5.4
no ip helper statistics vlan <vlan id> address <ipv4addr>    ! per vlan mode
ip helper agent-information <enable | disable>
ip helper agent-information <replace | keep | drop>
ip helper boot-up <enable | disable>
ip helper pxe-support <enable | disable>
```

Generic UDP Relay Service

```
ip udp relay {service <name> | port < number>}
ip udp relay {service <name> | port < number>} vlan [number]
ip udp relay no {service <name> | port < number>}
ip udp relay {service <name> | port < number>} no vlan [number }
```

In AOS7 and AOS8, it is still required to create the UDP port to forward either by well known service name or UDP port and then the destination VLANs or VLAN range."`ip udp relay no {service <name> | port <number>}`" deletes the entire associated destination VLANs configuration for that service or port.
In AOS7 and AOS8 the term "`service`" means a Generic UDP Relay Service which will be translated to an UDP port.

9.3. Basic troubleshooting

DHCP Relay Agent Service

Display statistics:

```
-> show ip helper statistics
Global Statistics :
    Reception From Client :
      Total Count =         21, Delta =         15
    Forw Delay Violation :
      Total Count =          0, Delta =          0
    Max Hops Violation :
      Total Count =          0, Delta =          0
    Agent Info Violation :
      Total Count =          0, Delta =          0
    Invalid Gateway IP :
      Total Count =          0, Delta =          0
Server Specific Statistics :
    From any Vlan to Server 192.168.10.254
        Tx Server :
          Total Count =         14, Delta =         14
        InvAgentInfoFromServer:
          Total Count =          0, Delta =          0
```

Clear statistics:

```
-> no ip helper statistics
-> no ip helper statistics global-only
-> no ip helper statistics server-only
```

Clear statistics in the standard mode:

```
-> no ip helper statistics address <ipv4addr>
```

Clear statistics in the per vlan mode:

```
-> no ip helper statistics vlan <vlan id> address <ipv4addr>
```

Generic UDP Relay Service

Use the "`show ip udp relay [service <name>| port [<number>]`" command to display UDP Relay configuration:

```
-> show ip udp relay
 Service Name         Port  Vlans
---------------------+-----+-----------------------------
DNS port              53
```

Use the "`show ip udp relay statistics [service <name>| port [<number>]`" command to display UDP Relay statistics:

```
-> show ip udp relay statistics
Port  Service        Pkts Recvd Pkts Sent  Dst Vlan
-----+--------------+----------+----------+--------
   53 DNS port                2
                                         2       10
```

Resetting statistics:

```
-> ip udp relay no statistics
```

DHCP Snooping Traffic Violation Statistics

DHCP Snooping traffic filtering/blocking statistics are kept per port. There are five counters:

- MAC Address violation counter. This counter is incremented when an DHCP packet is received on an untrusted interface, and the Ethernet source MAC address and the DHCP client hardware address do not match.
- DHCP Server packets violation counter. This counter is incremented when a DHCP packet from a DHCP server, such as a DHCPOFFER, DHCPACK, DHCPNAK, or DHCPLEASEQUERY packet, is received on an untrusted port.
- DHCP binding violation counter. This counter is incremented when the switch receives a DHCPRELEASE or DHCPDECLINE broadcast message that contains a MAC address in the DHCP snooping binding table, but the interface information in the binding table does not match the interface on which the message was received.
- DHCP Option 82 violation counter. This counter is incremented when a relay agent forwards a packet that includes option-82 information to an untrusted port.
- DHCP Relay Agent counter. This counter is incremented when a DHCP relay agent forwards a DHCP packet that includes a relay-agent IP address that is not 0.0.0.0.

Note: The above statistics violation counters are applicable for both switch-level and vlan-level DHCP Snooping. And they are only applicable when the port is in the "Client-Only" trust mode. When the port mode is change from "Client-Only" to "Blocked/Trusted", the counters are reset to 0

Logging to SWLOG

Following appids may be used for monitorig UDP Relay:

```
-> show configuration snapshot system
! System Service:
swlog appid udpRelay subapp all level debug3
swlog appid ipni subapp 15 level debug3
```

An example of UDP Relay related logs:

```
-> show log swlog | grep -E "udpRelay|udprelay"
<snap> swlogd: ipni udprelay debug2(7) port-chk: udp-dport 53
```

```
<snap> swlogd: ipni udprelay debug1(6) port-chk: match
<snap> swlogd: ipni udprelay debug3(8) ipni_is_udp_pkt_dhcp: udp-dport 53
<snap> swlogd: ipni udprelay debug2(7) pkt (len 98) sent to UDP-CMM
<snap> swlogd: udpRelay main debug1(6) [UDP Relay Debug]: IP ni Receive(slot 1).... len = 130
<snap> swlogd: udpRelay main debug1(6)  [UDP Relay Debug]: Processing pket from NI ....
<snap> swlogd: udpRelay main debug1(6)  UdpPort = 4529
<snap> swlogd: udpRelay main debug1(6)  Gport   = 47
<snap> swlogd: udpRelay main debug1(6)  VLAN    = 100
<snap> swlogd: udpRelay main debug1(6)  VRF     = 0
<snap> swlogd: udpRelay main debug1(6)  Pkt len = 98
<snap> swlogd: udpRelay main debug1(6) UDP Relay CMM: udpRelay_pktForwarding ...
<snap> swlogd: udpRelay main debug1(6) [dhcpSnoopingCheckLpsViolation:10637] Gport: 47
<snap> swlogd: udpRelay main debug3(8) dhcpSnoopingGetPortEntry: ifIndex 1048
<snap> swlogd: udpRelay main debug3(8) dhcpSnoopingGetPortEntry: ret 0xaa498
<snap> swlogd: udpRelay main debug1(6) udpRelay_pktForwarding: gport: 47, vid: 100, pkt len 102
vrf = 0, ifIndex = 1048
<snap> swlogd: udpRelay main debug1(6) ...packet get transfered to genericHandleRequest: Msg
type=0 VRF=0
<snap> swlogd: udpRelay main debug1(6)genericHandleRequest:from vlan = 100,sport = 4529, dport=53
<snap> swlogd: udpRelay main debug1(6)   Generic REQ: req recd on i/f addr = 0xc0a864fe
<snap> swlogd: ipni udprelay debug2(7) pkt (len 72) sent to UDP-CMM
<snap> swlogd: udpRelay main debug1(6)  [UDP Relay Debug]: IP ni Receive (slot 1).... len = 104
<snap> swlogd: udpRelay main debug1(6)  UdpPort = 4530
<snap> swlogd: udpRelay main debug1(6)  Pkt len = 72
<snap> swlogd: udpRelay main debug1(6)  udpRelay_pktForwarding: gport: 47, vid: 100, pkt len 76
vrf = 0, ifIndex = 1048
<snap> swlogd: udpRelay main debug1(6)genericHandleRequest:from vlan = 100, sport = 4530,dport=53
<snap> shasta swlogd: ipni udprelay debug2(7) port-chk: udp-dport 137
```

9.4. **Advanced troubleshooting**

DHCP Relay operation can be monitored using `"debug ip packet"`.

An example of a DHCP Renew (the client is connected to port 1/1/48, the server is connected to port 1/1/1):

```
-> debug ip packet protocol udp start timeout 60
-> 1 1 R 1/1/48 0007e9:1ff566->ffffff:ffffff IP 0.0.0.0->255.255.255.255 UDP 68,67
1 1 S UDP 0007e9:1ff566->ffffff:ffffff IP 0.0.0.0->255.255.255.255 UDP 68,67
1 C S 1/1/1 e8e732:ab17bd->00e0b1:f47b19 IP 192.168.10.253->192.168.10.254 UDP 67,67
1 1 R CMM1 e8e732:ab17bd->00e0b1:f47b19 IP 192.168.10.253->192.168.10.254 UDP 67,67
1 1 S 1/1/1 e8e732:ab17bd->00e0b1:f47b19 IP 192.168.10.253->192.168.10.254 UDP 67,67
1 1 S FLD e8e732:ab17bd->ffffff:ffffff ARP Request 192.168.100.254->192.168.100.100
1 1 R 1/1/1 00e0b1:f47b19->e8e732:ab17bd IP 192.168.10.254->192.168.100.254 UDP 67,67
1 1 S UDP 00e0b1:f47b19->e8e732:ab17bd IP 192.168.10.254->192.168.100.254 UDP 67,67
1 1 R UDP e8e732:ab17bd->0007e9:1ff566 IP 192.168.100.254->255.255.255.255 UDP 67,68
1 1 S 1/1/48 e8e732:ab17bd->0007e9:1ff566 IP 192.168.100.254->255.255.255.255 UDP 67,68
1 1 R 1/1/48 0007e9:1ff566->ffffff:ffffff IP 0.0.0.0->255.255.255.255 UDP 68,67
1 1 S UDP 0007e9:1ff566->ffffff:ffffff IP 0.0.0.0->255.255.255.255 UDP 68,67
1 C S 1/1/1 e8e732:ab17bd->00e0b1:f47b19 IP 192.168.10.253->192.168.10.254 UDP 67,67
1 1 R CMM1 e8e732:ab17bd->00e0b1:f47b19 IP 192.168.10.253->192.168.10.254 UDP 67,67
1 1 S 1/1/1 e8e732:ab17bd->00e0b1:f47b19 IP 192.168.10.253->192.168.10.254 UDP 67,67
1 1 R 1/1/1 00e0b1:f47b19->e8e732:ab17bd IP 192.168.10.254->192.168.100.254 UDP 67,67
1 1 S UDP 00e0b1:f47b19->e8e732:ab17bd IP 192.168.10.254->192.168.100.254 UDP 67,67
1 1 R UDP e8e732:ab17bd->0007e9:1ff566 IP 192.168.100.254->255.255.255.255 UDP 67,68
1 1 S 1/1/48 e8e732:ab17bd->0007e9:1ff566 IP 192.168.100.254->255.255.255.255 UDP 67,68
1 1 R 1/1/48 0007e9:1ff566->ffffff:ffffff ARP Request 192.168.100.100->192.168.100.100
1 1 R 1/1/48 0007e9:1ff566->ffffff:ffffff ARP Request 192.168.100.100->192.168.100.100
1 1 R 1/1/48 0007e9:1ff566->ffffff:ffffff ARP Request 192.168.100.100->192.168.100.100
```

An example of DHCP Release (the client is connected to port 1/1/48, the server is connected to port 1/1/1):

```
-> debug ip packet protocol udp start timeout 60
-> 1 1 R 1/1/48 0007e9:1ff566->e8e732:ab17bd IP 192.168.100.100->192.168.10.254 UDP 68,67
1 1 S UDP 0007e9:1ff566->e8e732:ab17bd IP 192.168.100.100->192.168.10.254 UDP 68,67
1 C S FLD e8e732:ab17bd->ffffff:ffffff ARP Request 192.168.10.253->192.168.10.254
1 1 R CMM e8e732:ab17bd->ffffff:ffffff ARP Request 192.168.10.253->192.168.10.254
1 1 S FLD e8e732:ab17bd->ffffff:ffffff ARP Request 192.168.10.253->192.168.10.254
1 1 R 1/1/1 00e0b1:f47b19->e8e732:ab17bd ARP Reply 192.168.10.254->192.168.10.253
```

```
1 C S 1/1/1 e8e732:ab17bd->00e0b1:f47b19 IP 192.168.10.253->192.168.10.254 UDP 67,67
1 1 R CMM1 e8e732:ab17bd->00e0b1:f47b19 IP 192.168.10.253->192.168.10.254 UDP 67,67
1 1 S 1/1/1 e8e732:ab17bd->00e0b1:f47b19 IP 192.168.10.253->192.168.10.254 UDP 67,67
```

9.5. **Troubleshooting in Maintenance Shell**

Warning: Maintenance Shell commands should only be used by Alcatel-Lucent personnel or under the direction of Alcatel-Lucent. Misuse or failure to follow procedures that use Maintenance Shell commands in this guide correctly can cause lengthy network down time and/or permanent damage to hardware.

Monitoring the number of DHCP messages receved on the Packet Driver level:

```
SHASTA #-> cat /proc/pktdrv | grep -E "Classified 24"
Classified 24       : 165679713        4
```

Monitoring software counters for IPv6 helper:

```
#-> telnet cmma 22012
dhcp6r> show proto
VRF default
Relay
  enabled=0 clientIfId=1 maxHops=32
  Stats:
    upstreamRx=0 downstreamRx=0 otherRx=0 disabledRx=0
    badLen=0 maxHopsExceeded=0 noLinkAddr=0 tooBig=0
    noAddress=0 invalidOption=0 missingOption=0
```

upstreamRx: Messages received from upstream, a DHCPv6 server or another relay agent. i.e. RELAY-REPLYs
downstreamRx: Messages received from downstream, either any of the client messages or a RELAY-FORWARD from another relay agent

There are a number of other debug commands, enter `'?'` or `'help'` at the dhcp6r> prompt for a list. It is highly recommended that only the `"show"` commands be used in the debug CLI. Debug CLI commands can change at any time (i.e. no guarantees on them working in the future if included in test scripts).

10. Troubleshooting QoS

Checklist

- Make sure that the "`log`" keyword in "`policy rule`" is realy needed - presense of this keyword is increasing TCAM usage by 1 entry per rule
- Use masks for MAC and IP addresses wherever possible
- Prefer actions with "`maximum bandwidth`" than "`cir`"
- Use manually optimized network group in place of build in "Switch"
- Specify source ports where applicable

Summary of the commands in this chapter is listed here:

```
show configuration snapshot vfc
show qos slice
show qos slice <slot/port>
tail -f vfc1.log
debug qos internal "slot 1 list 255 verbose"
d chg fp_port_field_sel
d chg fp_tcam
d chg fp_policy_table
d chg fp_meter_table
d chg fp_counter_table
```

10.1. Introduction

Trident hardware limitations

Number of meters and counters supported in the Trident's ICAP is 2048. Each meter bucket size can be programmed from 512B to 64KB. The granularity for the meter ranges from 8Kbps to 10Gbps. These 2K meters are globally located outside the IFP and grouped into 4 meter pools. Each meter pool has 256 meter pairs (containing odd and even meter).
Each port supports the egress shaping by using the leaky bucket. The meter bucket size can be programmed from 256B to 32KB. The granularity for the meter ranges from 8 Kbps to 10 Gbps.
Trident slices don't support double wide mode. It will support only the slice pairing mode for all the slices. In 7.1.1.R01, double wide mode is used for CPU-Q slice. Here CPU-Q slice entries are configured in single slice (7) in single wide mode.
Trident doesn't support WRED statistics for different colors at queue level. It will support drop statistics for each color at the port level.
The BCM56840Ax Errata says that in 640G device, there can be inaccuracies in egress port shaping mechanism. Accuracy of shaping can degrade with a function of the packet size number of active ports and specified shaping rate.In worst case this shaping rate can degrade by ~16%. However BCM56840 480G is unaffected by this issue. So Egress shaping in 640G device in full line rate can be unpredictable. From the software side we can create a set of Unit Tested result and share it but test has to fine tune the actual behavior.
OS6900 Platform uses Trident in the data path which supports 10 slices. In 10 slices, first 4 slices contain 128 entries per slice and the remaining 6 slices have 256 entries per slice, resulting in total of 2048 IFP entries.

Reserved slices

OS6860

Helix4 has 16 slices of 256 entries each:

- Slice 0 - Reserved for untrusted port entries/low priority (overridable) system slice
- Slice 1 - Reserved for IP protocol cpu priority entries
- Slices 14 & 15 - Reserved for high priority (non overridable) system slices

Features which require TCAM reservation:

- QoS Policies - can dynamically use all free slices based on policy configuration
- SIP snooping - will use between 1-4 slices based on a static tunable when enabled
- FIP snooping - reserves a single slice when enabled
- OpenFlow - Will reserve all free slices (2-13) when enabled (no other applications can be in use if it is enabled)
- AntiSpoofing - reserves a single slice when enabled
- Vlan Stacking / SPB SAPs - can dynamically use all free slices based on configuration
- *,G: - reserves a single slice when enabled
- DHCP Snooping - reserves a single slice on ASICs where DHCP snooping ports are enabled
- Deep Packet Inspection - reserves a single slice when enabled

No cache

For 'Switch' group policies it's recommended to use the 'no-cache' keyword in the action for the rule (it really should be part of the rule, but for historical reasons it's in the action). That will cause the policy to not be programmed into the hardware TCAM and only be matched in software. It is recommended to use AOS 7.3.1.643.R01 or later - given test traffic in the lab, show health shows no change in CPU utilization.
It's may be tricky to get right though, especially in the case there are rules with a lower precedence "deny all" and higher precedence "accept" policies. All policies that come after the first no-cache policy will need to be carefully checked to see if they also need to be no-cache, since if there's any overlap then they'll match first if they're in hardware.

Maximum bandwidth per port

There are two ways of limiting ingress banwidth:

```
qos port <chassis/slot/port> maximum ingress-bandwidth <bw> maximum depth <depth>
```

or

```
interfaces <chassis/slot/port> ingress-bandwidth mbps <bw> burst <burst>
interfaces <chassis/slot/port> ingress-bandwidth enable
```

Minimum and maximum bandwidth per queue

The following configuration from AOS 6:

```
qos port slot/port qn {minbw | maxbw} kbps
```

Can be replaced by:

```
-> show configuration snapshot vfc
! Virtual Flow Control:
qos qsp dcb "port11001" import qsp dcb "dcp-1"
qos qsp dcb "port11001" tc 1 min-bw 0 max-bw 20
qos qsp dcb "port11001" tc 2 max-bw 20
qos qsi port 1/1/1 qsp dcb "port11001"
```

QoS/ACL Design and Configuration

Introduction

QoS software on OmniSwitch 6900 provides a way to manipulate flows coming through the switch based on user configured policies such as ACLs, traffic prioritization, bandwidth shaping or traffic marking and mapping. ACLs are a specific type of QoS policy used for Layer 2, Layer 3/4, and multicast filtering. The below is intended to provide the QoS/ACL necessary information on the OmniSwitch 6900 series to successfully configure and design a QoS/ACL policy in your networks.

Examples provided in this document are taken from an OmniSwitch 6900 running 7.3.2.355.R01 with 26 ports which is a one NI (Network Interface) version. NI consists of a switching ASIC and physical ports.

Policy Condition and Action Guidelines

List of the Policy Conditions and Actions Available

Qos Conditions

Layer 1 Conditions

- Source / destination port
- Source / destination port group

Layer 2 Conditions

- Source MAC / MAC group
- Destination MAC / MAC group
- 802.1p
- Ethertype
- Source VLAN
- Destination VLAN (multicast rules only)
- Outer VLAN
- Inner VLAN

Layer 3 Conditions

- IP Protocol
- Source / destination IP
- Source / destination network group
- ToS
- DSCP
- ICMP code/type
- IPV6 Destination IP, Traffic, Next Header, Flow Label
- Multicast IP / Network Group

Layer 4 Conditions

- Source / destination TCP/UDP port
- Destination
- Service
- Service group
- ICMP type
- TCP Flags

IP Multicast

- For IGMP ACLs QoS Actions

Qos Actions

- ACL Drop
- Priority, specify the egress queue (0-7)
- Specify the maximum queue depth
- 802.1p stamping and mapping
- ToS stamping and mapping
- DSCP stamping and mapping
- Permanent gateway (Policy Based Routing)
- Maximum Bandwidth
- Port Redirection
- Port Disable

Policy Condition Combination Matrix

&	Layer 1	Layer 2	Layer 3	Layer 4	IP Multicast(IGMP)
Layer 1	All	All	All	All	Destination Only
Layer 2	All	All	All	Source VLAN and 802.1p only	Destination Only
Layer 3	All	All	All	All	Destination Only
Layer 4	All	Source VLAN and 802.1p only	All	All	None
IP Multicast(IGMP)	Destination Only	Destination Only	Destination Only	None	N/A

Policy Action Combination Matrix

&	Drop	Priority	Stamp/Map	Maximum Bandwidth	Redirect Port	Redirect Linkagg
Drop	N/A	No	No	No	No	No
Priority	No	N/A	Yes	Yes	Yes	Yes
Stamp/Map	No	Yes	N/A	Yes	Yes	Yes
Maximum Bandwidth	No	Yes	Yes	N/A	Yes	Yes
Redirect Port	No	Yes	Yes	Yes	N/A	No

Redirect Linkagg	No	Yes	Yes	Yes	No	N/A

Warning: Reflexive rules and NAT are not supported.

Understanding the QoS/ACL implementation on the OmniSwitch 6900

On the OmniSwitch 6900, QoS and ACL classification and actions are performed in hardware. QoS/ACL rules are a combination of conditions and actions. The OS6900 can classify, stamp and prioritize on Layer 2 through Layer 4 traffic simultaneously, whether bridging or routing.

Summary of guidelines to understand the QoS and TCAM usage and successfully configure policy rules on OS6900

The switching ASIC on each switching ASIC processes QoS and ACLs internally uses TCAM (Ternary Content Addressable Memory).

- The TCAM is divided into 14 slices, including 10 IFP slices and 4 EFP slices.
- IFP slices 0, 1, 2, and 3 can accommodate 128 rules, all other slices can accommodate 256 rules.
- IFP slices 0, 1, 2, 3, 8, and 9 are reserved for the system use. User policy rules aren’t configured in these slices.
- 4 IFP slices are available in User Policy Rules Space allowing 4 • 256 = 1024 rules.
- User policy rules are always in lower slices than the slice(s) allocated for Anti-spoofing, Ethernet-Service, DHCP IP Source Filtering.
- User policy cannot overwrite the sap-profile priority assignment and rate limiting.
- TCAM rules are programmed on every NI if the policy rule does not specify a source port. If policy is applied on a stack, rules without specified source port are configured across all units and their NIs.
- Once the slices are set with their parsing modes, a packet will be looked-up in parallel in all slices.
- When a match occurs in one slice, the parsing stop in that slice.
- The result of all matches among all slices in ANDed with the highest slice number actions having the higher precedence in case of similar actions with the following exceptions (for equal precedence and different slice number):
- A drop has always precedence over accept
- The smallest rate limiter is always enforced
- Policy Network / MAC / Map / Destination Slot-Port / Service Group used in a policy rule consume one TCAM rule for every entry in the group.
- Combining different policy groups in the policy rules consumes one TCAM rule for every possible combination of match between the groups.
- 8 TCP/UDP hardware ranges are available on the Packet Processor.
- TCP/UDP hardware ranges are being used when the range consumes more than 5 TCAM regular rules, using a TCP/UDP hardware range consumes 1 TCAM rule.
- If the same rule type requires more than 256 entries, a second slice (set with the same classification type is allocated).
- All rules with the same type (all Field Processing Selectors are set to the same mode) fall in the same slice (up to 256 entries).
- The rule precedence is based on the order in which the rule entry is entered or by defining the precedence in the rule. If precedence is not specified, rule entered first will have higher precedence.
- Change in precedence will automatically revisit all the slices and TCAM rule allocation.
- Efficient usage of the policy rule precedence allows the user to configure more rules and can avoid reaching the system limitation.
- Policy Types can be summarized.

TCAM allocation rules and practical examples

QoS starts allocating rules in IFP slice 7, and works towards IFP slice 4. Each slice is set up to look at a particular set of fields in a packet. If all the entries in a slice are used, or the slice can’t be programmed to accommodate all the fields needed in the condition, QoS moves on to the next lower slice.

Simple policy rules consuming only 1 TCAM entry

Single source IP
Configuration

```
policy condition c1 source IP 1.1.1.1
policy action a1 disposition accept
policy rule r1 condition c1 action a1
qos apply
```

TCAM Utilization
1 Rule is consumed on the NI

```
-> show qos slice
Slot/                  Ranges          Rules          Counters         Meters
Unit            Type Total/Free  CAM Total/Free      Total/Free      Total/Free
   1/(0)        IFP    32/32       0   128/128         128/128         128/128
                                   1   128/127         128/127         128/128
                                   2   128/125         128/125         128/128
                                   3   128/127         128/127         128/128
                                   4   256/256         256/256         256/256
                                   5   256/256         256/256         256/256
                                   6   256/256         256/256         256/256
                                   7   256/255         256/255         256/256
                                   8   256/255         256/254         256/254
                                   9   256/255         256/256         256/256
   1/(0)        EFP     0/0        0   256/256         256/256         256/256
                                   1   256/256         256/256         256/256
                                   2   256/256         256/256         256/256
                                   3   256/256         256/256         256/256
```

Single Source Network

Configuration

```
policy condition c1 source IP 1.1.1.0 mask 255.255.255.0
policy action a1 disposition accept
policy rule r1 condition c1 action a1
qos apply
```

TCAM Utilization

1 Rule is consumed on the NI

```
-> show qos slice
Slot/                  Ranges          Rules          Counters         Meters
Unit            Type Total/Free  CAM Total/Free      Total/Free      Total/Free
   1/(0)        IFP    32/32       0   128/128         128/128         128/128
                                   1   128/127         128/127         128/128
                                   2   128/125         128/125         128/128
                                   3   128/127         128/127         128/128
                                   4   256/256         256/256         256/256
                                   5   256/256         256/256         256/256
                                   6   256/256         256/256         256/256
                                   7   256/255         256/255         256/256
                                   8   256/255         256/254         256/254
                                   9   256/255         256/256         256/256
   1/(0)        EFP     0/0        0   256/256         256/256         256/256
                                   1   256/256         256/256         256/256
                                   2   256/256         256/256         256/256
                                   3   256/256         256/256         256/256
```

Mixing single source IP and destination IP in a policy condition will consume only one TCAM entry
Configuration

```
policy condition c1 source ip 1.1.1.1 destination ip 2.2.2.2
policy action a1 disposition accept
policy rule r1 condition c1 action a1
qos apply
```

TCAM Utilization
1 Rule is consumed on the NI

```
-> show qos slice
Slot/                  Ranges          Rules          Counters         Meters
Unit            Type Total/Free  CAM Total/Free      Total/Free      Total/Free
   1/(0)        IFP    32/32       0   128/128         128/128         128/128
                                   1   128/127         128/127         128/128
                                   2   128/125         128/125         128/128
                                   3   128/127         128/127         128/128
                                   4   256/256         256/256         256/256
                                   5   256/256         256/256         256/256
                                   6   256/256         256/256         256/256
                                   7   256/255         256/255         256/256
                                   8   256/255         256/254         256/254
                                   9   256/255         256/256         256/256
   1/(0)        EFP     0/0        0   256/256         256/256         256/256
                                   1   256/256         256/256         256/256
                                   2   256/256         256/256         256/256
                                   3   256/256         256/256         256/256
```

Policy conditions consuming multiple TCAM entries for a single policy rule

Example: Adding individual elements in a policy groups will increase the number of TCAM rules consumed

Configuration

```
policy network group g1 1.1.1.1 2.2.2.2 3.3.3.3
policy condition c1 source network group g1
policy action a1 disposition accept
policy rule r1 condition c1 action a1
qos apply
```

TCAM Utilization
This configuration consumes 1 TCAM rule for each entry in the network group on every NI, so 3 rules are consumed in total in slice 7 with this configuration. 253 rules in slice 7 are now available.

```
-> show qos slice</span>
Slot/                  Ranges          Rules          Counters         Meters</span>
Unit            Type Total/Free  CAM Total/Free      Total/Free      Total/Free</span>
   1/(0)        IFP    32/32       0   128/128         128/128         128/128
                                   1   128/127         128/127         128/128
                                   2   128/125         128/125         128/128
                                   3   128/127         128/127         128/128
                                   4   256/256         256/256         256/256
                                   5   256/256         256/256         256/256
                                   6   256/256         256/256         256/256
                                   7   256/253         256/253         256/256
                                   8   256/255         256/254         256/254
                                   9   256/255         256/256         256/256
   1/(0)        EFP     0/0        0   256/256         256/256         256/256
                                   1   256/256         256/256         256/256
                                   2   256/256         256/256         256/256
                                   3   256/256         256/256         256/256
```

Combining different policy groups in the policy rules can consume one TCAM rule for every possible combination of match between the groups

Configuration

```
policy network group g1 1.1.1.1 2.2.2.2 3.3.3.3
policy network group g2 4.4.4.4 5.5.5.5 6.6.6.6
policy condition c1 source network group g1 destination network group g2
policy action a1 disposition accept
policy rule r1 condition c1 action a1
qos apply
```

TCAM Utilization
This configuration consumes 9 TCAM rules = 3 Source IP • 3 Destination IP for every possible match between the group on the NI:

```
Source IP = 1.1.1.1 and Destination IP = 4.4.4.4
Source IP = 1.1.1.1 and Destination IP = 5.5.5.5
Source IP = 1.1.1.1 and Destination IP = 6.6.6.6
Source IP = 2.2.2.2 and Destination IP = 4.4.4.4
Source IP = 2.2.2.2 and Destination IP = 5.5.5.5
Source IP = 2.2.2.2 and Destination IP = 6.6.6.6
Source IP = 3.3.3.3 and Destination IP = 4.4.4.4
Source IP = 3.3.3.3 and Destination IP = 5.5.5.5
Source IP = 3.3.3.3 and Destination IP = 6.6.6.6
```

```
-> show qos slice
Slot/                  Ranges          Rules          Counters         Meters
Unit            Type Total/Free  CAM Total/Free      Total/Free      Total/Free
   1/(0)        IFP    32/32       0   128/128         128/128         128/128
                                   1   128/127         128/127         128/128
                                   2   128/125         128/125         128/128
                                   3   128/127         128/127         128/128
                                   4   256/256         256/256         256/256
                                   5   256/256         256/256         256/256
                                   6   256/256         256/256         256/256
                                   7   256/247         256/247         256/256
                                   8   256/255         256/254         256/254
                                   9   256/255         256/256         256/256
   1/(0)        EFP     0/0        0   256/256         256/256         256/256
                                   1   256/256         256/256         256/256
                                   2   256/256         256/256         256/256
                                   3   256/256         256/256         256/256
```

A single policy condition consumes 2 TCAM rules due to slice pairs

Rules of consumption
In AOS7, Source IP conditions and destination IP conditions use paired TCAM slices.

Example: Source IPv6 condition consume 1 TCAM rule in a slice pair

Configuration

```
policy condition c1 source vlan 559 source ipv6 2001:4cd0:bc00:2570:1::1 destination tcp-port 80
policy action a1
policy rule r1 condition c1 action a1
qos apply
```

TCAM Utilization
1 rule consumed in slice pair 6_7.

```
-> show qos slice
Slot/                  Ranges          Rules          Counters         Meters
Unit            Type Total/Free  CAM Total/Free      Total/Free      Total/Free
 1/1/(0)        IFP    32/32       0   128/128         128/128         128/128
                                   1   128/127         128/127         128/128
                                   2   128/125         128/125         128/128
                                   3   128/127         128/127         128/128
                                   4   256/256         256/256         256/256
                                   5   256/256         256/256         256/256
                                   6   256/255         256/255         256/256
                                   7   256/255         256/256         256/256
                                   8   256/255         256/254         256/254
```

```
                                    9   256/255         256/256         256/256
 1/1/(0)        EFP      0/0        0   256/256         256/256         256/256
                                    1   256/256         256/256         256/256
                                    2   256/256         256/256         256/256
                                    3   256/256         256/256         256/256</pre>
```

Combination of policy rules leading to multiple TCAM slices consumption

Rules of consumption

The number of configurable policies can be reduced due to the TCAM slice allocation. As explained before, there are 4 TCAM slices available (remaining 4 slices are reserved) with 256 entries on each slice for user configuration. Usually, while creating policy rules the system allocates the TCAM entries on the same slice until the 256 entries are used then it moves to the next slice and so on up to exhaustion of all available entries.

Warning: Source slot-port rules and destination slot-port rules cannot use the same slice. L2 rules and L4 rules cannot use the same slice. Souce IPv6 and destination IPv6 rules cannot use the same slice.

Example of a mix of rules consuming different slices

Configuration

```
policy condition c1 source vlan 559 source ipv6 2001:4cd0:bc00:2570:1::1 destination tcp-port 80
policy condition c2 source vlan 519 destination ipv6 2001:4cd0:bc00:2570:1::1 source tcp-port 80
policy action a1
policy rule r1 condition c1 action a1
policy rule r2 condition c2 action a1
qos apply
```

TCAM Utilization
Source IPv6 rules and destination IPv6 rules cannot use the same slice pair. FPSs (Field Processing Selectors) in the slice pairs are configured differently. A single slice pair cannot hold both rules r1 and r2, as a result these rules are programmed in different slice pairs and each consumes 1 TCAM rule.

```
-> show qos slice
Slot/                 Ranges          Rules          Counters         Meters
Unit            Type Total/Free  CAM Total/Free      Total/Free      Total/Free
 1/1/(0)        IFP    32/32      0   128/128         128/128         128/128
                                  1   128/127         128/127         128/128
                                  2   128/125         128/125         128/128
                                  3   128/127         128/127         128/128
                                  4   256/255         256/255         256/256
                                  5   256/255         256/256         256/256
                                  6   256/255         256/255         256/256
                                  7   256/255         256/256         256/256
                                  8   256/255         256/254         256/254
                                  9   256/255         256/256         256/256
 1/1/(0)        EFP     0/0       0   256/256         256/256         256/256
                                  1   256/256         256/256         256/256
                                  2   256/256         256/256         256/256
                                  3   256/256         256/256         256/256
```

TCAM exhausted when hitting the system limitation, the importance of the policy rule precedence
The number of rules available on the system can be exhausted if switch hits the system limitation (rules programmed on different slices or policy network group rules are not optimized).

Example 1: TCAM exhausted when the default precedence is used
Source IPv6 rules and destination IPv6 rules cannot use the same slice pair: The following combination of source IPv6 rules and destination IPv6 rules forces the system to program each entry in different slice pair because the user is letting the system using the default precedence order.

Warning: The rule precedence is based on the order in which the rule entry is entered or by defining the precedence in the rule. If precedence is not specified, rule entered first will have higher precedence.

Configuration

```
policy condition c1 source vlan 559 source ipv6 2001:4cd0:bc00:2570:1::1 destination tcp-port 80
policy condition c2 source vlan 519 destination ipv6 2001:4cd0:bc00:2570:1::1 source tcp-port 80
policy condition c3 source vlan 559 source ipv6 2001:4cd0:bc00:2570:1::2 destination tcp-port 80
policy action a1
policy rule r1 condition c1 action a1
policy rule r2 condition c2 action a1
policy rule r3 condition c3 action a1
ERROR: Out of TCAM processors on 1/0(0)
```

TCAM Utilization

Rule 3 cannot be configured on the system because rules 1 and 2 are already using every 4 slices available (2 slice pairs), so rule 3 cannot use the same slice pair as rule 2. The switch returns an error stating that the system is out of TCAM processors.

Example 2: TCAM exhausted because of a manually misconfigured precedence order

The following mix of source IPv6 rules and destination IPv6 rules will cause the TCAM to allocate 1 entry per slice pair because of the precedence order. The 3rd rule cannot be programmed into the hardware because no slice is available.

Configuration

```
policy condition c1 source vlan 559 source ipv6 2001:4cd0:bc00:2570:1::1 destination tcp-port 80
policy condition c2 source vlan 519 destination ipv6 2001:4cd0:bc00:2570:1::1 source tcp-port 80
policy condition c3 source vlan 559 source ipv6 2001:4cd0:bc00:2570:1::2 destination tcp-port 80
policy action a1
policy rule r1 condition c1 action a1 precedence 100
policy rule r2 condition c2 action a1 precedence 110
policy rule r3 condition c3 action a1 precedence 120
ERROR: Out of TCAM processors on 1/0(0)
```

TCAM Utilization

Rule 3 cannot be configured on the system because rules 1 and 2 are already using all 4 slices available (2 slice pairs), so rule 3 cannot use the same slice pair as rule 2. The switch returns an error stating that the system is out of TCAM processors.

Example 3: Similar to example 1 and 2 with an optimized and working configuration – Efficient usage of the Precedence

Configuration

```
policy condition c1 source vlan 559 source ipv6 2001:4cd0:bc00:2570:1::1 destination tcp-port 80
policy condition c2 source vlan 519 destination ipv6 2001:4cd0:bc00:2570:1::1 source tcp-port 80
policy condition c3 source vlan 559 source ipv6 2001:4cd0:bc00:2570:1::2 destination tcp-port 80
policy action a1
policy rule r1 condition c1 action a1 precedence 110
policy rule r2 condition c2 action a1 precedence 100
policy rule r3 condition c3 action a1 precedence 120
qos apply
```

TCAM Utilization

This configuration will use slice pair 6_7 for the source IPv6 rules and slice pair 4_5 for the destination IPv6 rule.

This configuration will accomplish the same purpose as example 1 and example 2 while consuming only 4 slices while the 2 previous examples were consuming too many slices and were not supported on the system.

```
-> show qos slice
Slot/                  Ranges          Rules          Counters         Meters
Unit            Type Total/Free  CAM Total/Free      Total/Free       Total/Free
 1/1/(0)        IFP    32/32       0   128/128         128/128          128/128
                                   1   128/127         128/127          128/128
                                   2   128/125         128/125          128/128
```

```
                                        3   128/127         128/127         128/128
                                        4   256/255         256/255         256/256
                                        5   256/255         256/256         256/256
                                        6   256/254         256/254         256/256
                                        7   256/254         256/256         256/256
                                        8   256/255         256/254         256/254
                                        9   256/255         256/256         256/256
 1/1/(0)        EFP     0/0             0   256/256         256/256         256/256
                                        1   256/256         256/256         256/256
                                        2   256/256         256/256         256/256
                                        3   256/256         256/256         256/256
```

Layer 4, TCP/UDP ports, service groups and port ranges

Warning: A single TCP/UDP port rule will consume one TCAM rule. TCP/UDP port ranges consume one or multiple TCAM rules entries depending on the range. 8 Hardware TCP/UDP ranges are available and automatically used instead of the regular TCAM rules when the range is supposed to consume 6 or more than 6 rules.

The Classifier Processor on the ASIC has a separate table with a capacity of 8 TCP/UDP port ranges per TCAM. Each port range will consume one TCAM entry and we can have 8 rules which use the TCP/UDP port range table. However, the user can configure more than 8 TCP/UDP port ranges, additional TCP/UDP port ranges consuming more than 5 TCAM rules are programmed to the TCAM using multiple TCAM entries. Hardware TCP/UDP port ranges are only allocated for TCP/UDP port ranges that require 6 or more than 6 regular TCAM entries. TCP/UDP port ranges that can be programmed directly to the TCAM using less than 6 TCAM entries will not consume a hardware range table entry.

Understanding the TCAM rule consumption for Layer 4 rules and TCP/UDP port ranges

Source and destination ports are 2 bytes long fields in the TCP/UDP headers. To understand the rules consumption you need to convert the TCP/UDP value from decimal to binary and check which mask to apply to fit to the port or port range, depending on the range a single may cover multiple values:

```
Single Port: 80 (decimal) -> (binary)              80 =       00000000 01010000
                                                   value       00000000 01010000
                                                    mask      11111111 11111111
```

Consumes 1 TCAM rule

```
Port Range: 2-3 (decimal) -> (binary)         2 =  00000000 00000010
                                              3 =  00000000 00000011
                                             value00000000 00000010
                                              mask 11111111 11111110
```

Consumes 1 TCAM rule

In this example, ports 2 and 3 can use the same mask. The first 15 bits for port 2 and port 3 are identical.

```
Port Range: 2-4 (decimal) -> (binary)         2 =  00000000 00000010
                                              3 =  00000000 00000011
                                              4 =  00000000 00000100
```

Consumes 2 TCAM rules
One single rule is used to perform a match when the port number is equal 2 or 3 since both values share a common mask:

```
TCAM rule 1                      value1  00000000 00000010 (match port 2-3)
                                  mask1  11111111 11111110 (mask port 2-3)
```

One additional TCAM rule is used to match when the port number equals to 4:

```
TCAM rule 2                      value2  00000000 00000100 (match port 4)
                                  mask2  11111111 11111111 (mask port 4)
```

In this example, it is not possible to find a single mask to cover port 2, 3 and 4. As a result the switch will consume 2 TCAM rules.

TCAM entries consumption examples for TCP/UDP port range

- Source TCP port 0-10 consumes 3 TCAM rules
- Source TCP port 1-10 consumes 5 TCAM rules
- Source TCP port 0-32 consumes 2 TCAM rules
- Source TCP port 1-32 is supposed to consume 6 TCAM, so 1 hardware TCP/UCP range is used in combination with only 1 TCAM rule
- Source TCP port 0-65535 consumes 1 TCAM rule
- Source TCP port 1-65535 is supposed to consume 16 TCAM rules, so 1 hardware TCP/UCP range is used in combination with only 1 TCAM rule

Example 1: Layer 4 – a single port

Configuration

```
policy service http destination tcp 80
policy condition c1 service http
policy action a1 disposition accept
policy rule r1 condition c1 action a1
qos apply
```

Consumes 1 TCAM rule on the NI

Explanation

```
Single Port: 80 (decimal) -> (binary)        80 =    00000000 01010000
                                           value    00000000 01010000
                                            mask    11111111 11111111
```

In order to match this value the rule has to do an exact match for port 80

Example 2: Layer 4 – a port range

Configuration

```
policy condition c1  source tcp 1-2
policy action a1 disposition accept
policy rule r1 condition c1 action a1
qos apply
```

Will consume 2 TCAM rules on the NI

Explanation

```
Port Range: 1-2 (decimal) -> (binary)        1 =     00000000 00000001
                                             2 =     00000000 00000010
```

One single rule is used to match when the port number is equal 1:

```
TCAM rule 1                           value1       00000000 00000001 (match port 1)
                                       mask1       11111111 11111111 (full mask)
```

Another rule is used to match when the port number is equal 2:

```
TCAM rule 2                         value2  00000000 00000010 (match port 2)
                                     mask2  11111111 11111111 (full mask)
```

Example 3: Layer 4 – a port range

Configuration

```
policy condition c1 source tcp 1-10
policy action a1 disposition accept
policy rule r1 condition c1 action a1
qos apply
```

Will consume 5 TCAM rules on the NI

Explanation

In order to understand this example, you have to convert each TCP port decimal value to its binary value, see bellow:

```
1     (decimal) -> (binary)    0000 0001      <-       Rule 1
2     (decimal) -> (binary)    0000 0010      <-       Rule 2
3     (decimal) -> (binary)    0000 0011
4     (decimal) -> (binary)    0000 0100      <-       Rule 3
5     (decimal) -> (binary)    0000 0101
6     (decimal) -> (binary)    0000 0110
7     (decimal) -> (binary)    0000 0111
8     (decimal) -> (binary)    0000 1000      <-       Rule 4
9     (decimal) -> (binary)    0000 1001
10    (decimal) -> (binary)    0000 1010      <-       Rule 5</pre>
```

```
-> show qos slice
Slot/               Ranges          Rules          Counters         Meters
Unit            Type  Total/Free  CAM Total/Free      Total/Free      Total/Free
   1/(0)        IFP    32/32       0   128/128         128/128         128/128
                                   1   128/127         128/127         128/128
                                   2   128/125         128/125         128/128
                                   3   128/127         128/127         128/128
                                   4   256/256         256/256         256/256
                                   5   256/256         256/256         256/256
                                   6   256/256         256/256         256/256
                                   7   256/251         256/251         256/256
                                   8   256/255         256/254         256/254
                                   9   256/255         256/256         256/256
   1/(0)        EFP     0/0        0   256/256         256/256         256/256
                                   1   256/256         256/256         256/256
                                   2   256/256         256/256         256/256
                                   3   256/256         256/256         256/256
```

Example 4: Hardware TCP/UDP port range is used if more than 6 TCAM are supposed to be consumed by the port range

Configuration

```
policy condition c1 source tcp 1-32
policy action a1 disposition accept
policy rule r1 condition c1 action a1
qos apply
```

TCAM Utilization

This rule would consume 6 TCAM rules, so the system automatically uses 1 TCP/UDP Hardware Range and 1 TCAM rule on the NI.

Explanation

When the policy rule is consuming more than 5 regular TCAM rules the system automatically allocates a dedicate hardware TCP/UDP range for this policy so that it only consumes 1 TCAM rule, this allows the system to save TCAM rules for other rules.

```
-> show qos slice
```

```
Slot/                  Ranges          Rules          Counters         Meters
Unit            Type Total/Free  CAM Total/Free      Total/Free      Total/Free
   1/(0)        IFP    32/31       0   128/128         128/128         128/128
                                   1   128/127         128/127         128/128
                                   2   128/125         128/125         128/128
                                   3   128/127         128/127         128/128
                                   4   256/256         256/256         256/256
                                   5   256/256         256/256         256/256
                                   6   256/256         256/256         256/256
                                   7   256/255         256/255         256/256
                                   8   256/255         256/254         256/254
                                   9   256/255         256/256         256/256
   1/(0)        EFP     0/0        0   256/256         256/256         256/256
                                   1   256/256         256/256         256/256
                                   2   256/256         256/256         256/256
                                   3   256/256         256/256         256/256
```

Example of a more complex rule

Configuration

```
policy service SERV1 destination tcp 80
policy service SERV2 destination tcp 21
policy service group SERV_GRP SERV1 SERV2
policy port group Dest_Port_group 1/1 1/12 1/20
policy network group SRCNet1 12.12.12.1 12.12.12.2 12.12.12.3 12.12.12.4 12.12.12.5
policy network group DSTNet1 22.21.21.1 22.21.21.2 22.21.21.3 22.21.21.4 22.21.21.5 22.21.21.10
policy condition CC1 destination port group Dest_Port_group source network group SRCNet1
destination network group DSTNet1 service
group SERV_GRP
policy action AA1 disposition drop
policy rule RR1 condition CC1  action AA1
qos apply
```

TCAM Utilization

180 = (2 services • 3 egress ports • 5 source networks • 6 destination networks)
180 rules are consumed on each NI in the system

```
-> show qos slice
Slot/                  Ranges          Rules          Counters         Meters
Unit            Type Total/Free  CAM Total/Free      Total/Free      Total/Free
   1/(0)        IFP    32/31       0   128/128         128/128         128/128
                                   1   128/127         128/127         128/128
                                   2   128/125         128/125         128/128
                                   3   128/127         128/127         128/128
                                   4   256/256         256/256         256/256
                                   5   256/256         256/256         256/256
                                   6   256/76          256/76          256/256
                                   7   256/255         256/255         256/256
                                   8   256/255         256/254         256/254
                                   9   256/255         256/256         256/256
   1/(0)        EFP     0/0        0   256/256         256/256         256/256
                                   1   256/256         256/256         256/256
                                   2   256/256         256/256         256/256
                                   3   256/256         256/256         256/256
```

10.2. Basic Troubleshooting

- Following message is displayed when a configured rule would exceed the system capacity:
  ERROR: Out of TCAM processors on 1/0(0)
- To determine how many rules and masks are being used by the system use "`show qos slice`" or "`show qos slice slot/port`" commands (available from the CLI).
- In case of heavy traffic matching a rule with "`log`" enable, it is possible to observe packet loss

10.3. **Advanced troubleshooting**

Dislay QoS rules configured in TCAM

Notes:

- In a VC chassis id needs to be specified, for example: debug qos internal "chassis 1 slot 1 list 1 verbose"
- In AOS 7.3.3.R01 use "slot 0/1 list 0 verbose" or "chassis 0 slot 1 list 0 verbose"

All lists:

```
-> debug qos internal "slot 1 list 255 verbose"
               Entry U Slice CIDU CIDL MIDU MIDL TCAM     Count[+]             Green[+]
Red[+]          NotGreen[+]
List 0 empty
List 1: 19 entries set up
       McastARP(  17) 0     8 1543    -    -    - 1636         0[0        ]         0[0        ]
0[0        ]         0[0
       McastARP(  17) 0     9    -    -    -    - 1892         0[0        ]         0[0        ]
0[0        ]         0[0
     ISIS_BPDU1(  22) 0     8    - 1536    -    - 1590     10738[10738    ]         0[0        ]
0[0        ]         0[0
...
```

List 0 (user rules):

```
-> debug qos internal "slot 1 list 0 verbose"
               Entry U Slice CIDU CIDL MIDU MIDL TCAM     Count[+]             Green[+]
Red[+]          NotGreen[+]
List 0: 1 entries set up
         Policy(   0) 0     7    - 1280    -    - 1280         0[0        ]         0[0        ]
0[0        ]         0[0
```

List 1 (all copy to CPU):

```
-> debug qos internal "slot 1 list 1 verbose"
               Entry U Slice CIDU CIDL MIDU MIDL TCAM     Count[+]             Green[+]
Red[+]          NotGreen[+]
List 1: 20 entries set up
     HgMcastARP(  16) 0    -1    -    -    0    0   -2         0[0        ]         0[0        ]
0[0        ]         0[0
       McastARP(  17) 0     8 1543    -    -    - 1636         0[0        ]         0[0        ]
0[0        ]         0[0
       McastARP(  17) 0     9    -    -    -    - 1892         0[0        ]         0[0        ]
0[0        ]         0[0
     ISIS_BPDU1(  22) 0     8    - 1536    -    - 1590         0[0        ]         0[0        ]
0[0        ]         0[0
     ISIS_BPDU1(  22) 0     9    -    -    -    - 1846         0[0        ]         0[0        ]
0[0        ]         0[0
     ISIS_BPDU2(  23) 0     8 1537    -    -    - 1591         0[0        ]         0[0        ]
0[0        ]         0[0
     ISIS_BPDU2(  23) 0     9    -    -    -    - 1847         0[0        ]         0[0        ]
0[0        ]         0[0
     ISIS_BPDU3(  24) 0     8    - 1538    -    - 1592         0[0        ]         0[0        ]
0[0        ]         0[0
     ISIS_BPDU3(  24) 0     9    -    -    -    - 1848         0[0        ]         0[0        ]
0[0        ]         0[0
      IPMS_IGMP(  50) 0     8 1539    -    -    - 1638       576[0        ]         0[0        ]
0[0        ]         0[0
      IPMS_IGMP(  50) 0     9    -    -    -    - 1894       576[0        ]         0[0        ]
0[0        ]         0[0
 IPMS_V4Control(  51) 0     8    - 1540    -    - 1639      1829[5        ]         0[0        ]
0[0        ]         0[0
 IPMS_V4Control(  51) 0     9    -    -    -    - 1895      1829[0        ]         0[0        ]
0[0        ]         0[0
    IPMS_V4Data(  52) 0     8 1541    -    -    - 1640         0[0        ]         0[0        ]
0[0        ]         0[0
    IPMS_V4Data(  52) 0     9    -    -    -    - 1896         0[0        ]         0[0        ]
0[0        ]         0[0
```

```
IPMS_V4Resolved(  53) 0     8    - 1542    -    - 1641         0[0        ]          0[0        ]
0[0        ]         0[0
IPMS_V4Resolved(  53) 0     9    -    -    -    - 1897         0[0        ]          0[0        ]
0[0        ]         0[0
     ETHOAM_SYS(  62) 0     8    - 1562    -    - 1741         0[0        ]          0[0        ]
0[0        ]         0[0
     ETHOAM_SYS(  62) 0     9    -    -    -    - 1997         0[0        ]          0[0        ]
0[0        ]         0[0
802.1ab Regular( 102) 0     8 1547    -    -    - 1586         8[1        ]          0[0        ]
0[0        ]         0[0
802.1ab Regular( 102) 0     9    -    -    -    - 1842         8[0        ]          0[0        ]
0[0        ]         0[0
   amap Regular( 103) 0     8    - 1546    -    - 1587         3[1        ]          0[0        ]
0[0        ]         0[0
   amap Regular( 103) 0     9    -    -    -    - 1843         3[0        ]          0[0        ]
0[0        ]         0[0
802.3ad Regular( 104) 0     8    - 1548    -    - 1588       257[47       ]          0[0        ]
0[0        ]         0[0
802.3ad Regular( 104) 0     9    -    -    -    - 1844       257[0        ]          0[0        ]
0[0        ]         0[0
 802.1x Regular( 105) 0     8 1549    -    -    - 1589         0[0        ]          0[0        ]
0[0        ]         0[0
 802.1x Regular( 105) 0     9    -    -    -    - 1845         0[0        ]          0[0        ]
0[0        ]         0[0
   BPDU Regular( 106) 0     8    - 1550    -    - 1584     27022[67       ]          0[0        ]
0[0        ]         0[0
   BPDU Regular( 106) 0     9    -    -    -    - 1840     27022[0        ]          0[0        ]
0[0        ]         0[0
           MPLS( 120) 0     8    - 1544    -    - 1543         0[0        ]          0[0        ]
0[0        ]         0[0
           MPLS( 120) 0     9    -    -    -    - 1799         0[0        ]          0[0        ]
0[0        ]         0[0
      srcsldrop( 128) 0     8 1551    -    -    - 1536         0[0        ]          0[0        ]
0[0        ]         0[0
      srcsldrop( 128) 0     9    -    -    -    - 1792         0[0        ]          0[0        ]
0[0        ]         0[0
Static Mac Move( 136) 0     8 1553 1552    0    1 1750         0[0        ]          0[0        ]
0[0        ]         0[0
Static Mac Move( 136) 0     9    -    -    -    - 2006         0[0        ]          0[0        ]
0[0        ]         0[0
            MIM( 143) 0     8 1545    -    -    - 1537         0[0        ]          0[0        ]
0[0        ]         0[0
            MIM( 143) 0     9    -    -    -    - 1793         0[0        ]          0[0        ]
0[0        ]         0[0
     LINKOAMSAA( 161) 0     8 1561    -    -    - 1742         0[0        ]          0[0        ]
0[0        ]         0[0
     LINKOAMSAA( 161) 0     9    -    -    -    - 1998         0[0        ]          0[0        ]
0[0        ]         0[0
```

List 2:

```
-> debug qos internal "slot 1 list 2 verbose"
               Entry U Slice CIDU CIDL MIDU MIDL TCAM     Count[+]             Green[+]
Red[+]          NotGreen[+]
List 2: 6 entries set up
       IPMS_MLD(  78) 0     2  257    -    -    -  258         0[0        ]          0[0        ]
0[0        ]         0[0
 IPMS_V6Control(  81) 0     2    -  258    -    -  259         0[0        ]          0[0        ]
0[0        ]         0[0
    IPMS_V6Data(  82) 0     2  259    -    -    -  260         0[0        ]          0[0        ]
0[0        ]         0[0
IPMS_V6Resolved(  83) 0     2    -  260    -    -  261         0[0        ]          0[0        ]
0[0        ]         0[0
      MPLSTrust( 124) 0     2  261    -    -    -  256         0[0        ]          0[0        ]
0[0        ]         0[0
       MIMTrust( 144) 0     2    -  262    -    -  257         0[0        ]          0[0        ]
0[0        ]         0[0
```

List 7:

```
-> debug qos internal "slot 1 list 7 verbose"
              Entry U Slice CIDU CIDL MIDU MIDL TCAM     Count[+]             Green[+]
Red[+]          NotGreen[+]
List 7: 66 entries set up
      PortTrust(   1) 0     2  263    -    -    -  263         0[0        ]         0[0        ]
0[0        ]         0[0
      PortTrust(   2) 0     2    -  264    -    -  264         0[0        ]         0[0        ]
0[0        ]         0[0
      PortTrust(   3) 0     2  265    -    -    -  265         0[0        ]         0[0        ]
0[0        ]         0[0
      PortTrust(   4) 0     2    -  266    -    -  266         0[0        ]         0[0        ]
0[0        ]         0[0
      PortTrust(   5) 0     2  267    -    -    -  267         0[0        ]         0[0        ]
0[0        ]         0[0
      PortTrust(   6) 0     2    -  268    -    -  268         0[0        ]         0[0        ]
0[0        ]         0[0
...
```

List 9 (phones auto QoS):

```
-> debug qos internal "slot 1 list 8 verbose"
              Entry U Slice CIDU CIDL MIDU MIDL TCAM     Count[+]             Green[+]
Red[+]          NotGreen[+]
List 8: 7 entries set up
      AutoPhone(   0) 0     8    - 1554    -    - 1679         0[0        ]         0[0        ]
0[0        ]         0[0
      AutoPhone(   0) 0     9    -    -    -    - 1935         0[0        ]         0[0        ]
0[0        ]         0[0
      AutoPhone(   1) 0     8 1555    -    -    - 1680         0[0        ]         0[0        ]
0[0        ]         0[0
      AutoPhone(   1) 0     9    -    -    -    - 1936         0[0        ]         0[0        ]
0[0        ]         0[0
      AutoPhone(   2) 0     8    - 1556    -    - 1681         0[0        ]         0[0        ]
0[0        ]         0[0
      AutoPhone(   2) 0     9    -    -    -    - 1937         0[0        ]         0[0        ]
0[0        ]         0[0
      AutoPhone(   3) 0     8 1557    -    -    - 1682         0[0        ]         0[0        ]
0[0        ]         0[0
...
```

List 12 (L3 features, all copy to CPU):

```
-> debug qos internal "slot 1 list 12 verbose"
              Entry U Slice CIDU CIDL MIDU MIDL TCAM     Count[+]             Green[+]
Red[+]          NotGreen[+]
List 12: 16 entries set up
           ospf(  64) 0     3    -  384    -    -  448      1838[1838     ]         0[0        ]
0[0        ]         0[0
           vrrp(  65) 0     3  385    -    -    -  449         0[0        ]         0[0        ]
0[0        ]         0[0
           icmp(  66) 0     3    -  386    -    -  450         0[0        ]         0[0        ]
0[0        ]         0[0
            rip(  67) 0     3  387    -    -    -  451         0[0        ]         0[0        ]
0[0        ]         0[0
         bgpsrc(  68) 0     3    -  388    -    -  452      1766[1766     ]         0[0        ]
0[0        ]         0[0
         bgpdst(  69) 0     3  389    -    -    -  453         0[0        ]         0[0        ]
0[0        ]         0[0
        bfdecho(  70) 0     3    -  390    -    -  454         0[0        ]         0[0        ]
0[0        ]         0[0
       bfdreply(  71) 0     3  391    -    -    -  455         0[0        ]         0[0        ]
0[0        ]         0[0
         telnet(  72) 0     3    -  392    -    -  456         0[0        ]         0[0        ]
0[0        ]         0[0
            ssh(  73) 0     3  393    -    -    -  457         0[0        ]         0[0        ]
0[0        ]         0[0
           http(  74) 0     3    -  394    -    -  458         0[0        ]         0[0        ]
0[0        ]         0[0
           snmp(  75) 0     3  395    -    -    -  459         0[0        ]         0[0        ]
0[0        ]         0[0
```

```
            arp(  76) 0      3    -  396    -    -  460       115[115      ]         0[0        ]
0[0        ]         0[0
          ripng(  77) 0      3  397    -    -    -  461         0[0        ]         0[0        ]
0[0        ]         0[0
            pim(  78) 0      3    -  398    -    -  462         0[0        ]         0[0        ]
0[0        ]         0[0
          mcipc(  82) 0      3  399    -    -    -  463    183182[183182   ]         0[0        ]
0[0        ]         0[0
```

10.4. Troubleshooting in the Maintenance Shell

Warning: Maintenance Shell commands should only be used by Alcatel-Lucent personnel or under the direction of Alcatel-Lucent. Misuse or failure to follow procedures that use Maintenance Shell commands in this guide correctly can cause lengthy network down time and/or permanent damage to hardware.

VFC Troubleshooting

Swlog is extremely important to trace any VFC-related issues. All VFC-related swlog is stored in the /var/log directory called "vfc1.log". Note that the swlog is cleared every time the OmniSwitch OS6900 or 10K is rebooted.

For example if VFC error messages appeared on the console and you want to see more details as to what happened during that time, login as super user "su", cd to the /var/log directory, open the vfc.log file, and search for that particular timestamp when the error event happened. To troubleshoot VFC in real-time while having the console connection active, open a telnet session and do the following as indicated below:

```
#-> cd /var/log
#-> ls
fd1.log ipms.log lag.log mcm.log vfc1.log vm.log vstk.log wtmp
#-> tail -f vfc1.log
19 16:49:04 - dbg: [vfcInitSlotProfile:249] Create Transaction buffer vfcTxBuf[0
19 16:49:04 - dbg: [vfcInitSlotProfile:254] zNI 0, profiling done
19 16:49:04 - dbg: [vfcInitSlotProfile:249] Create Transaction buffer vfcTxBuf[1
19 16:49:04 - dbg: [vfcInitSlotProfile:254] zNI 1, profiling done
19 16:49:04 - dbg: [vfcConnectToCS:52] VFC Connect to CS
19 16:49:04 - dbg: [vfcConnectToPM:602] VFC Connected to PM
19 16:49:04 - dbg: [main:407] ==generated MIB database==
19 16:49:04 - dbg: [main:410] VFC cslib_unblock
19 16:49:04 - dbg: [vfcMainLoop:301] vfcMainLoop
19 16:49:04 - dbg: [vfcHandleMipMsg:380] Queuing the MIP message
19 16:49:04 - dbg: [getQsapRangeFromIfIndex:1920] Before EOIC: received PORT qsa
19 16:49:04 - dbg: [getQsapRangeFromIfIndex:1920] Before EOIC: received PORT qsa
19 16:49:04 - dbg: [vfc_qsap_control_prop:2185] Invalid QSI 719 16:49:04 - dbg:
19 16:49:04 - dbg: [vfcHandleMipMsg:380] Queuing the MIP message
19 16:49:04 - dbg: [getQsapRangeFromIfIndex:1907] Before EOIC: received LAG qsap
19 16:49:04 - dbg: [vfcHandleMipMsg:380] Queuing the MIP message
19 16:49:04 - dbg: [vfcMipEoicFunction:268] EOIC received
19 16:50:15 - dbg: [vfcAddNewConnection:256] New connection: 127.2.65.1:37985, S
19 16:50:15 - dbg: [vfcHandleIncomingMsg:140] NI 0 connected after NI DOWN, sock
19 16:50:15 - dbg: [vfcHandleIncomingMsg:147] RX VFC_MSG_HELLO zNi 0 BOOTUP
19 16:50:15 - dbg: [vfcPMEventsRegister:575] Port Manager Registrations Done
```

10.5. Troubleshooting in bShell

Display Field Processing Selector configuration (in AOS the configuration is always the same for all ports):

```
BCM.0> d chg fp_port_field_sel
FP_PORT_FIELD_SEL.ipipe0[0]:
<SLICE9_S_TYPE_SEL=2,SLICE9_F3=6,SLICE9_8_PAIRING=1,SLICE8_S_TYPE_SEL=2,SLICE8_F3=3,SLICE8_F2=5,
SLICE8_F1=1,SLICE7_S_TYPE_SEL=1,SLICE7_F2=5,SLICE7_F1=5,SLICE6_S_TYPE_SEL=1,SLICE5_S_TYPE_SEL=1,S
LICE4_S_TYPE_SEL=1,
SLICE3_S_TYPE_SEL=2,SLICE3_F3=5,SLICE3_F1=1,SLICE2_S_TYPE_SEL=1,SLICE2_F3=6,SLICE2_F2=4,SLICE2_F1
=6,SLICE1_S_TYPE_SEL=1,
SLICE1_F1=2,SLICE0_S_TYPE_SEL=1,SLICE0_F1=2>
FP_PORT_FIELD_SEL.ipipe0[1]:
<SLICE9_S_TYPE_SEL=2,SLICE9_F3=6,SLICE9_8_PAIRING=1,SLICE8_S_TYPE_SEL=2,SLICE8_F3=3,SLICE8_F2=5,
SLICE8_F1=1,SLICE7_S_TYPE_SEL=1,SLICE7_F2=5,SLICE7_F1=5,SLICE6_S_TYPE_SEL=1,SLICE5_S_TYPE_SEL
=1,SLICE4_S_TYPE_SEL=1,
SLICE3_S_TYPE_SEL=2,SLICE3_F3=5,SLICE3_F1=1,SLICE2_S_TYPE_SEL=1,SLICE2_F3=6,SLICE2_F2=4,SLICE2_F1
=6,SLICE1_S_TYPE_SEL=1,
SLICE1_F1=2,SLICE0_S_TYPE_SEL=1,SLICE0_F1=2>
…
```

Display configured configured in TCAM (the entry number corresponds to the TCAM column from << debug qos internal `"slot 1 list 255 verbose"` >> output):

```
BCM.0> d chg fp_tcam
FP_TCAM.ipipe0[256]: <VALID=3,PAIRING_F1_MASK=0x18000000,PAIRING_F1=0x18000000,
 MASK=0x600000000000000000000000000000000000000000000000000,
 KEY=0x600000000000000000000000000000000000000000000000000,F1_MASK=0x18000000,F1=0x18000000>
FP_TCAM.ipipe0[257]: <VALID=3,PAIRING_F1_MASK=0x20000000,PAIRING_F1=0x20000000,
 MASK=0x800000000000000000000000000000000000000000000000000,
 KEY=0x800000000000000000000000000000000000000000000000000,F1_MASK=0x20000000,F1=0x20000000>
FP_TCAM.ipipe0[258]: <VALID=3,PAIRING_FIXED_MASK=0x44,PAIRING_F3_MASK=0x18,PAIRING_F3=0x18,
PAIRING_F2_MASK=0xff000000000000000003fc00000000ff,PAIRING_F2=0xff000000000000000000e80000000001,
 MASK=0x2200000000003fc00000000000000000ff000000003fc00000000c00,
 KEY=0x3fc000000000000000003a0000000000400000000c00,FIXED_MASK=0x11,F3_MASK=0x18,F3=0x18,
 F2_MASK=0xff000000000000000003fc00000000ff,F2=0xff000000000000000000e80000000001>
FP_TCAM.ipipe0[259]: <VALID=3,PAIRING_FIXED_MASK=0x44,PAIRING_F3_MASK=0x18,PAIRING_F3=0x18,
PAIRING_F2_MASK=0xffffffffffffffff0000000000000000,PAIRING_F2=0xff020000000000000000000000000000,
MASK=0x22000000000003fffffffffffffffc000000000000000000000000c00,KEY=0x3fc080000000000000000000000
00000000000000c00,
FIXED_MASK=0x11,F3_MASK=0x18,F3=0x18,F2_MASK=0xffffffffffffffff0000000000000000,F2=0xff0200000000
00000000000000000000>
FP_TCAM.ipipe0[260]: <VALID=3,PAIRING_FIXED_MASK=0x44,PAIRING_F3_MASK=0x18,PAIRING_F3=0x18,
PAIRING_F2_MASK=0xff000000000000000000000000000000,PAIRING_F2=0xff000000000000000000000000000000,
PAIRING_F1_MASK=0x1e000000,PAIRING_F1=0xc000000,MASK=0x2200078000003fc000000000000000000000000000
00000000000c00,
 KEY=0x30000003fc00000000000000000000000000000000000000c00,FIXED_MASK=0x11,F3_MASK=0x18,F3=0x18,
F2_MASK=0xff000000000000000000000000000000,F2=0xff000000000000000000000000000000,F1_MASK=0x1e0000
00,F1=0xc000000>
FP_TCAM.ipipe0[261]: <VALID=3,PAIRING_FIXED_MASK=0x44,PAIRING_F3_MASK=0x18,PAIRING_F3=0x18,
PAIRING_F2_MASK=0xff000000000000000000000000000000,PAIRING_F2=0xff000000000000000000000000000000,
PAIRING_F1_MASK=0x1e000000,PAIRING_F1=0xe000000,MASK=0x2200078000003fc000000000000000000000000000
00000000000c00,
 KEY=0x38000003fc00000000000000000000000000000000000000c00,FIXED_MASK=0x11,F3_MASK=0x18,F3=0x18,
F2_MASK=0xff000000000000000000000000000000,F2=0xff000000000000000000000000000000,F1_MASK=0x1e0000
00,F1=0xe000000>
FP_TCAM.ipipe0[263]:
<VALID=3,PAIRING_FIXED_MASK=4,MASK=0x200000000000000000000000000000000000000000000000000000000,
 FIXED_MASK=1>
FP_TCAM.ipipe0[264]:
<VALID=3,PAIRING_FIXED_MASK=4,MASK=0x200000000000000000000000000000000000000000000000000000000,
 FIXED_MASK=1>
 ...
```

Display rules configured in TCAM (the entry number corresponds to the TCAM column from << debug qos internal `"slot 1 list 255 verbose"` >> output):

```
BCM.0> d chg fp_policy_table
FP_POLICY_TABLE.ipipe0[256]: <COUNTER_MODE=0x38,COUNTER_INDEX=2>
FP_POLICY_TABLE.ipipe0[257]: <EVEN_PARITY=1,COUNTER_MODE=7,COUNTER_INDEX=3>
FP_POLICY_TABLE.ipipe0[258]: <EVEN_PARITY=1,CPU_COS=0xe,COUNTER_MODE=0x38,CHANGE_CPU_COS=1>
FP_POLICY_TABLE.ipipe0[259]: <G_COPY_TO_CPU=1,EVEN_PARITY=1,COUNTER_MODE=7,COUNTER_INDEX=1>
FP_POLICY_TABLE.ipipe0[260]:
<Y_COPY_TO_CPU=3,R_COPY_TO_CPU=3,G_COPY_TO_CPU=3,COUNTER_MODE=0x38,COUNTER_INDEX=1>
FP_POLICY_TABLE.ipipe0[261]:
<Y_COPY_TO_CPU=3,R_COPY_TO_CPU=3,G_COPY_TO_CPU=3,COUNTER_MODE=7,COUNTER_INDEX=2>
FP_POLICY_TABLE.ipipe0[263]:  <Y_CHANGE_PKT_PRI=5,Y_CHANGE_D
SCP=3,Y_CHANGE_COS_OR_INT_PRI=5,R_CHANGE_PKT_PRI=5,
 R_CHANGE_D
SCP=3,R_CHANGE_COS_OR_INT_PRI=5,G_CHANGE_PKT_PRI=5,G_CHANGE_DSCP_TOS=3,G_CHANGE_COS_OR_INT_PRI=5>
```

```
FP_POLICY_TABLE.ipipe0[264]:
<Y_CHANGE_PKT_PRI=5,Y_CHANGE_DSCP=3,Y_CHANGE_COS_OR_INT_PRI=5,R_CHANGE_PKT_PRI=5,R_CHANGE_DSCP=3,
 R_CHANGE_COS_OR_INT_PRI=5,G_CHANGE_PKT_PRI=5,G_CHANGE_DSCP_TOS=3,G_CHANGE_COS_OR_INT_PRI=5>
FP_POLICY_TABLE.ipipe0[265]:
<Y_CHANGE_PKT_PRI=5,Y_CHANGE_DSCP=3,Y_CHANGE_COS_OR_INT_PRI=5,R_CHANGE_PKT_PRI=5,R_CHANGE_DSCP=3,
 R_CHANGE_COS_OR_INT_PRI=5,G_CHANGE_PKT_PRI=5,G_CHANGE_DSCP_TOS=3,G_CHANGE_COS_OR_INT_PRI=5>
...
```

Display meters (rate limiting) configured in TCAM (the entry number does NOT correspond to the TCAM column from, it corresponds to meter configured in FP_POLICY_TABLE):

```
BCM.0> d chg fp_meter_table
FP_METER_TABLE.ipipe0[0]: <REFRESHCOUNT=2,METER_GRAN=3,BUCKETSIZE=5,BUCKETCOUNT=0x50000>
FP_METER_TABLE.ipipe0[1]: <REFRESHCOUNT=2,METER_GRAN=3,BUCKETSIZE=5,BUCKETCOUNT=0x50000>
FP_METER_TABLE.ipipe0[2]: <BUCKETSIZE=1,BUCKETCOUNT=0xffff>
FP_METER_TABLE.ipipe0[3]: <BUCKETSIZE=1,BUCKETCOUNT=0x3ffffff>
```

Display counter configured in TCAM (the entry number does NOT correspond to the TCAM column from, it corresponds to counter configured in FP_POLICY_TABLE):

```
BCM.0> d chg fp_counter_table
FP_COUNTER_TABLE.ipipe0[399]: <PACKET_COUNTER=0x1bd48e,BYTE_COUNTER=0x1ff1f2b9>
FP_COUNTER_TABLE.ipipe0[1548]: <PACKET_COUNTER=0x1a5e,BYTE_COUNTER=0xd2f00>
```

11. Troubleshooting RIP

Summary of the commands in this chapter is listed here:

show ip rip interface
show ip redist rip
show ip rip
show ip rip peer
show ip rip routes
show ip route-map
show log swlog

Verify the required parameters for a RIP interface using the **show ip rip interface** command.

```
-> show ip rip interface "vlan_1"
Interface IP Name                       = vlan_1,
Interface IP Address                    = 192.168.6.2,
IP Interface Number (VLANId)            = 0,
Interface Admin status                  = disabled,
IP Interface Status                     = disabled,
Interface Config Ingress Route Map Name = ,
Interface Config Egress Route Map Name  = ,
Interface Config AuthType               = None,
Interface Config AuthKey Length         = 0,
Interface Config Send-Version           = v2,
Interface Config Receive-Version        = both,
Interface Config Default Metric         = 1,
Received Packets                        = 0,
Received Bad Packets                    = 0,
Received Bad Routes                     = 0,
Sent Updates                            = 0
```

This interface can be configured for RIP v 1 or RIP v 2. Now, the RIP interface must be enabled using the **ip rip interface** command.

```
-> ip rip interface "vlan_1" admin-state enable
-> show ip rip interface
       Interface             Intf Admin   IP Intf      Updates
         Name         vlan     status     status     sent/recv(bad)
---------------------+------+-----------+-----------+---------------
vlan_1              0      enabled     disabled    0/0(0)
```

The interface is enabled. Verify that local interface redistribution is enabled using the **show ip route-map** command.

```
-> show ip route-map
Route Maps: configured: 11 max: 200
Route Map: LOCAL4_RIP_1 Sequence Number: 1 Action permit
match ip-address 102.0.0.0/8 redist-control all-subnets deny
set metric 1 effect none

-> show ip route-map local_map_1
Route Map: local_map_1 Sequence Number: 1 Action permit
match ip-address 102.0.0.0/8 redist-control all-subnets deny
set metric 1 effect none
```

Verify that RIP is enabled globally and redistribution is also enabled, using the **show ip rip** command.

```
-> show ip rip
Status                     = Enabled,
Number of routes           = 0,
Number of prefixes         = 0,
Host Route Support         = Enabled,
Route Tag                  = 0,
Update interval            = 30,
Invalid interval           = 180,
Garbage interval           = 120,
Holddown interval          = 0,
Forced Hold-Down Timer     = 0
```

Now, verify if the peer relationship is established between the two routers using the **show ip rip peer** command.

```
-> show ip rip peer
     Total    Bad    Bad                Secs since
IP Address          Recvd   Packets Routes Version last update
----------------+--------+-------+------+-------+-----------
102.100.0.26       21773 0      0       2           17
102.101.0.26       21768 0      0       2           10
102.102.0.6        21760 0      0       2           27
102.102.0.26       21758 0      0       2           3
```

The above command output shows the number of updates received as well as the time since the last update. If the peer relationship is not formed, then the next thing to look for will be the other router to check if it is setup correctly.

Now, look at the routing table for RIP protocol, using the **show ip rip routes** command.

```
-> show ip rip routes
Legends: State: A = Active, H = Holddown, G = Garbage
Destination           Gateway           State Metric Proto
------------------+-----------------+----+------+------
105.0.0.0/8           +102.100.0.26      A    2      Rip
+102.101.0.26                             A    2      Rip
+102.102.0.26                             A    2      Rip
105.4.0.0/16          +11.102.15.1       A    1      Redist
105.12.0.0/16         +11.102.15.1       A    1      Redist
105.13.0.0/16          +11.102.15.1       A    1      Redist
105.21.0.0/16          +11.102.15.1       A    1      Redist
105.31.0.0/16          +11.102.15.1       A    1      Redist
192.168.0.0/24        +102.100.0.26      A    3      Rip
+102.101.0.26                             A    3      Rip
+102.102.0.26                             A    3      Rip
192.168.1.0/24        +102.100.0.26      A    3      Rip
+102.101.0.26                             A    3      Rip
+102.102.0.26                             A    3      Rip
192.168.2.0/24        +102.100.0.26      A    3      Rip
+102.101.0.26                             A    3      Rip
+102.102.0.26                             A    3      Rip
```

Next, clear the switch log using the **swlog clear** command and then wait for a few seconds and then run the **show log swlog** command.

```
-> show log swlog
Displaying file contents for '/flash/swlog2.log'
FILEID: fileName[/flash/swlog2.log], endPtr[60], configSize[500000], mode[2]
Displaying file contents for '/flash/swlog1.log'
FILEID: fileName[/flash/swlog1.log], endPtr[1433], configSize[500000], mode[1]
Time Stamp                Application    Level    Log Message
------------------------+--------------+-------+--------------------------------
TUE JUN 03 14:23:53 2008         SYSTEM    info Switch Logging cleared by command.
File Size=1000000 bytes
TUE JUN 03 14:24:00 2008            DRC    info tRip::ripRecv:Received packet from
102.102.0.26
```

```
TUE JUN 03 14:24:00 2008         DRC    info tRip::ripRecv: Rx: RESP ver=v2
src=102.102.0.26 inIf=102.102.0.122 port=520 tupl
TUE JUN 03 14:24:00 2008         DRC    info es=25 len=504
TUE JUN 03 14:24:00 2008         DRC    info tRip::ripRecv:Received packet from
102.102.0.26
TUE JUN 03 14:24:00 2008         DRC    info tRip::ripRecv: Rx: RESP ver=v2
src=102.102.0.26 inIf=102.102.0.122 port=520 tupl
TUE JUN 03 14:24:00 2008         DRC    info es=25 len=504
TUE JUN 03 14:24:00 2008         DRC    info tRip::ripRecv:Received packet from
102.102.0.26
TUE JUN 03 14:24:00 2008         DRC    info tRip::ripRecv: Rx: RESP ver=v2
src=102.102.0.26 inIf=102.102.0.122 port=520 tupl
TUE JUN 03 14:24:00 2008         DRC    info es=25 len=504
TUE JUN 03 14:24:00 2008         DRC    info tRip::ripRecv:Received packet from
102.102.0.26
TUE JUN 03 14:24:00 2008         DRC    info tRip::ripRecv: Rx: RESP ver=v2
src=102.102.0.26 inIf=102.102.0.122 port=520 tupl
```

12. Troubleshooting OSPF

Checklist

- Make sure that neighbors use the same MTU size

Note:
OSPF maximum packet size is hardcoded to 4 KB (to make sure that 8 KB out buffer is not fully used), but OSPF can still negotiate greater MTU size.

Summary of the commands in this chapter is listed here:

```
show configuration snapshot ospf ip
show ip ospf
show ip ospf neighbor
show ip ospf interface
show ip ospf interface <interface>
show ip ospf area
show ip ospf area <area id>
show ip ospf lsdb
show vrf
```

12.1. Supported debug variables

Note: areamaxintfs default is 200 interfaces per area

```
debug ip ospf set noloopback0 1
debug ip ospf set nostubloopback0 1
debug ip ospf set subsecond 1
debug ip ospf set bfdsubsecond 1
debug ip ospf set areamaxintfs <n>
```

OSPF 150ms link convergence made it in the GA build for 7.1.1.R01.
You have to enable it as follows:

```
debug ip ospf set subsecond 1
```

Further enhancement has been done to extend this feature in combination with BFD.
From Build 7.1.1.1673.R01 onwards (not sure if this is post-GA or not), if you need the same behavior (i.e. sub-second reconvergence) on BFD events in OSPF, you can enable:

```
debug ip ospf set bfdsubsecond 1
```

12.2. Planned and unplanned Virtual Chassis takeover

This technical document explains how OSPF graceful restart (unplanned) feature can be used on AOS7 Virtual Chassis to achieve sub-second convergence during Virtual Chassis takeover.

Overview on OSPF task in AOS7 VC
According to current AOS architecture, the control/management functions are performed by CMM and data forwarding functions are performed by NI. OSPF task runs in control plane building the forwarding information which will be used by the data plane for data forwarding. OSPF task functions in centralized mode

which means, an active OSPF process running in the primary CMM of the Master chassis controls data forwarding in all the NIs of the system.
During the system start up, OSPF is loaded (task spawned) in all the CMMs of the system, which includes the primary and secondary CMM of all the chassis in the system. However, OSPF will be activated ONLY on the primary CMM of the Master chassis. Active OSPF process enables the OSPF interfaces, sends Hello messages, discover neighboring routers, elect Designated Router (DR) and exchange link-state advertisements (LSAs). Once the LSA exchanges are completed, OSPF calculates the Shortest Path First (SPF) table and instructs IPRM to install the routes into all the NIs of the system. The neighbor router information and SPF table information will NOT be synced with OSPF running on other CMMs in the system. This is because OSPF minimizes the possibility of routing loops and/or black holes caused by lack of database synchronization between the Master and Slave chassis. OSPF process on other CMMs completes the initialization and waits for takeover message from Chassis Supervisor. It will not send/receive any protocol messages.

OSPF process during Virtual Chassis Takeover
When the Master chassis is reset or powered down, the Slave chassis takeover the control functions. During the takeover process, the chassis supervisor in the primary CMM of the Slave chassis sends takeover message to its OSPF task. On receiving the takeover message, the OSPF task on the primary CMM of the Slave chassis will be activated. The OSPF neighbor table and LSA database is rebuilt in the Slave chassis. The forwarding tables in the NI will remain intact throughout the takeover process intentionally to allow continuous forwarding of traffic across CMM takeover. However traffic forwarding is disrupted briefly during takeover. This undesired behavior is due to the following reason. When the adjacencies are formed with the neighboring routers, the sequence numbers used in protocol packets (DB Descriptor packets) are not retained across takeover causing the neighboring router to reset the adjacencies. This resetting of OSPF adjacencies results in neighboring routers flushing their forwarding table entries in NI causing traffic disruption.

OSPF Graceful Restart (Unplanned)
To overcome the traffic disruption due to adjacency reset during takeover, OSPF graceful restart feature is implemented. OSPF takeover could be either planned or unplanned. Since AOS7 supports only unplanned graceful restart feature, this document discuss only about unplanned graceful restart feature. Unplanned OSPF restart could be due Master chassis powered down or process crash in Master CMM. When the OSPF task on slave chassis receives takeover, it checks if the graceful restart feature is enabled. If yes, then OSPF enters graceful restart mode. On entering the graceful restart, OSPF on the restarting router first sends Graceful LSA Update message to the neighboring routers on the enabled OSPF interface. On receiving the Graceful LSA Update message, the neighboring router enters into helper mode, in which it will NOT reset the adjacency due to sequence number mismatch in protocol packets. The neighboring router continues to advertise the LSA of restarting router until the restarting router forms FULL adjacency. Once the restarting router forms FULL adjacency with its neighboring router, it sends Graceful LSA to terminate the graceful restart period.
Requirements for supporting graceful restart:

- The neighbor relationship status between the restarting router and neighbor router should be in “FULL” state in the neighbor router for processing the Graceful LSA Update from restarting router. If for any reason the neighbor goes down before the Graceful LSA Update message, then the neighbor router simply discard the LSA resulting in OSPF adjacency restart (flushes the forwarding table) when out of sequence protocol packets are received.
- There should not be any OSPF topology change in the network during the graceful restart period. If the neighboring router detects any OSPF network topology changes, then it updates the SPF table and resets the forwarding table in NI.
- Graceful LSA Update message should be sent out first before the Hello packet to the neighbor, to avoid adjacency reset due to OSPF state mismatch in Hello packet.

Sub-second convergence during Virtual Chassis Takeover
To achieve sub-second convergence during VC takeover, the requirements for OSPF graceful restart should be met.

- The OSPF Hello timer and Dead interval timer plays a key role in achieving sub-second convergence using graceful restart.
- These timers should be based around the time taken between the last Hello packet sent by the deceased Master chassis and the first Hello packet sent by the Slave chassis post takeover.
- This directly depends on the time taken by the Slave chassis to detect the Master chassis failure and time taken for the IP interfaces to come UP on the Slave chassis.
- Considering this dependency, the OSPF Hello timer and Dead interval timer MUST be non-aggressive.
- Having aggressive value for these timers can result in adjacency break down between the restarting router and neighboring routers before the Graceful LSA Update message was sent.
- This results in failure to achieve sub-second convergence during VC takeover.
- Default timer values for Hello timer (10 seconds) and Dead Interval timer (40 seconds) are recommended for OSPF Graceful Restart Support.

Significance of BFD during Virtual Chassis Takeover

The BFD feature helps OSPF for sub-second convergence when there is a fault in the bidirectional path between the adjacent routers. This is achieved by establishing BFD sessions (simple Hello mechanism with short intervals) between neighboring routers. The BFD session runs in the NI. When the NI application detects a failure in the BFD session, it gets communicated to the CMM application which in turn informs the registered protocol application to take necessary action.
In the case of a Virtual Chassis setup, the NI application in the Slave chassis will be UP and maintain the BFD session between the neighboring routers. This way the BFD sessions are not broken and there will not be any system/link failure reported to OSPF protocol application. Due to this, there is no significance for BFD configuration in OSPF takeover process.

12.3. Minimum working configuration

```
-> show configuration snapshot ospf ip
! IP:
ip interface "vlan1" address 192.168.1.1 mask 255.255.255.0 vlan 1
! OSPF:
ip load ospf
ip ospf area 0.0.0.0
ip ospf interface "vlan1"
ip ospf interface "vlan1" area 0.0.0.0
ip ospf interface "vlan1" admin-state enable
ip ospf admin-state enable
```

12.4. Basic troubleshooting

Displays the OSPF status and general configuration parameters:

```
-> show ip ospf
Router Id                        = 192.168.1.1,
OSPF Version Number              = 2,
Admin Status                     = Enabled,
Area Border Router ?             = No,
AS Border Router Status          = Disabled,
Route Tag                        = 0,
SPF Hold  Time (in seconds)      = 10,
SPF Delay  Time (in seconds)     = 5,
MTU Checking                     = Disabled,
# of Routes                      = 1,
# of AS-External LSAs            = 0,
# of self-originated LSAs        = 1,
# of LSAs received               = 2,
External LSDB Limit              = -1,
Exit Overflow Interval           = 0,
# of SPF calculations done       = 3,
# of Incr SPF calculations done  = 0,
# of Init State Nbrs             = 0,
# of 2-Way State Nbrs            = 0,
# of Exchange State Nbrs         = 0,
# of Full State Nbrs             = 1,
# of attached areas              = 2,
# of Active areas                = 1,
# of Transit areas               = 0,
# of attached NSSAs              = 0,
Default Route Origination        = none,
Default Route Metric-Type/Metric = type2 / 1,
BFD Status                       = Disabled
```

Displays information on OSPF non-virtual neighbor routers:

```
-> show ip ospf neighbor
  IP Address        Area Id          Router Id      Vlan   State  Type
----------------+----------------+----------------+------+-------+--------
192.168.1.2      0.0.0.0          192.168.1.2      1        Full  Dynamic
```

Displays information on OSPF non-virtual neighbor routers (detailed output):

```
-> show ip ospf neighbor 192.168.1.2
Neighbor's IP Address                = 192.168.1.2,
Neighbor's Router Id                 = 192.168.1.2,
Neighbor's Area Id                   = 0.0.0.0,
Neighbor's DR Address                = 192.168.1.2,
Neighbor's BDR Address               = 192.168.1.1,
Neighbor's Priority/Eligibility      = 1,
Neighbor's State                     = Full,
Hello Suppressed ?                   = No,
Neighbor's type                      = Dynamic,
# of State Events                    = 6,
Mode                                 = Slave,
MD5 Sequence Number                  = 0,
Time since Last Hello                = 4 sec,
# of Outstanding LS Requests         = 0,
# of Outstanding LS Acknowledgements = 0,
# of Outstanding LS Retransmissions  = 0,
Restart Helper Status                = notHelping,
Restart Age (in seconds)             = 0 sec,
Last Restart Helper Exit Reason      = None
```

Displays OSPF interface information:

```
-> show ip ospf interface
    Interface                DR          Backup DR      Admin    Oper             BFD
      Name                 Address         Address       Status  Status  State    Status
---------------------+----------------+----------------+--------+------+-------+-----------
vlan1                 192.168.1.2      192.168.1.1      enabled   up    BDR     disabled
```

Displays OSPF interface information (detailed output):

```
-> show ip ospf interface vlan1
Interface IP Name                     = vlan1,
VLAN Id                               = 1,
Interface IP Address                  = 192.168.1.1,
Interface IP Mask                     = 255.255.255.0,
Admin Status                          = Enabled,
Operational Status                    = Up,
OSPF Interface State                  = BDR,
Interface Type                        = Broadcast,
Area Id                               = 0.0.0.0,
Designated Router IP Address          = 192.168.1.2,
Designated Router RouterId            = 192.168.1.2,
Backup Designated Router IP Address   = 192.168.1.1,
Backup Designated Router RouterId     = 192.168.1.1,
MTU  (bytes)                          = 1500,
Metric Cost                           = 1,
Priority                              = 1,
Hello Interval (seconds)              = 10,
Transit Delay (seconds)               = 1,
Retrans Interval (seconds)            = 5,
Dead Interval (seconds)               = 40,
Poll Interval (seconds)               = 120,
Link Type                             = Broadcast,
Authentication Type                   = none,
# of Events                           = 26,
# of Init State Neighbors             = 0,
# of 2-Way State Neighbors            = 0,
# of Exchange State Neighbors         = 0,
# of Full State Neighbors             = 1,
BFD status                            = Disabled,
DR-Only Option for BFD                = Disabled
```

Displays all OSPF areas:

```
-> show ip ospf area
    Area Id      AdminStatus      Type       OperStatus
---------------+-------------+-------------+------------
0.0.0.0           enabled      normal       up
```

Displays a specified OSPF area information:

```
-> show ip ospf area 0.0.0.0
Area Identifier                          = 0.0.0.0,
Admin Status                             = Enabled,
Operational Status                       = Up,
Area Type                                = normal,
Area Summary                             = Enabled,
Time since last SPF Run                  = 00h:09m:12s,
# of Area Border Routers known           = 0,
# of AS Border Routers known             = 0,
# of Active Virtual Links                = 0,
# of LSAs in area                        = 3,
# of SPF Calculations done               = 3,
# of Incremental SPF Calculations done   = 0,
# of Neighbors in Init State             = 0,
# of Neighbors in 2-Way State            = 0,
# of Neighbors in Exchange State         = 0,
# of Neighbors in Full State             = 1,
# of Interfaces attached                 = 2,
Attached Interfaces                      = vlan1, vlan2
```

Displays LSAs in the Link State Database associated with each area:

```
-> show ip ospf lsdb
    Area Id        Type        LS Id        Orig Router-Id     SeqNo      Age
----------------+-------+----------------+----------------+------------+-----
0.0.0.0          rtr     192.168.1.1      192.168.1.1      0x80000003  584
0.0.0.0          rtr     192.168.1.2      192.168.1.2      0x80000004  581
0.0.0.0          net     192.168.1.2      192.168.1.2      0x80000002  580
```

12.5. **Advanced troubleshooting**

Enabling SWLOGs logs per VRF
The applicable protocols have switch logging application names with an appended number that indicates the VRF identifier ("vrfid") of the VRF in which the protocol is running. The default VRF always corresponds to "vrfid" zero. So the switch logging application-name for ospf in the default VRF is "ospf_0". To figure out which "vrfid" corresponds to a given non-default VRF name, one must examine the "ps" output in the su shell. An example output from a switch that is running OSPF in three VRFs:

```
-> show vrf
Virtual Routers     Profile Protocols
--------------------+-------+-------------------
```

```
default              default RIP OSPF BGP
a                    max     OSPF VRRP
b                    max     OSPF VRRP
Total Number of Virtual Routers: 3
-> su
Entering maintenance shell. Type 'exit' when you are done.
RUSHMORE #-> ps | grep [o]spf
2386 root     /bin/ospf
2671 root     /bin/ospf --vrfid 2 --vrfname b
2678 root     /bin/ospf --vrfid 1 --vrfname a
```

In the filtered "ps" output shown above, we see three processes corresponding to the OSPF instances running in each VRF. The OSPF task _without_ the "vrfid" and "vrfname" arguments is the one running in the default VRF. Each protocol process running in a non-default VRF will have "vrfid" and "vrfname" arguments. Those arguments tell us how to map VRF names to vrfids. Above, you can see that "vrfname b" is on the same line as "vrfid 2". So to control OSPF switch logging in VRF "b", we use switch logging application name "ospf_2". Similarly, to control OSPF switch logging in VRF "a", we use switch logging application name "ospf_1". {{Note|It's a coincidence that VRF name "a" happens to be vrfid 1 and vrf name "b" happens to be vrfid 2. The association between VRF names and vrfids is often less straightforward than that.}

Logging to SWLOG

All OSPF switch logging is emitted at level "debug2". If logging subapp is set to anything less than debug2, no log messages in the corresponding category will be emitted. Available subapps:

Subbapp ID	Subbapp name	Description
1	error	Error messages only. Error messages provide information of program faults.
2	warning	Warning messages only.
3	recv	Messages for packets received by OSPF only.
4	send	Messages for packets sent by OSPF only.
5	flood	Messages for the flooding of Link State Advertisements (LSAs) in OSPF only.
6	spf	Messages for OSPF’s Shortest Path First (SPF) calculations only.
7	lsdb	Messages for OSPF’s Link State Database (LSDB) related operations only.
8	rdb	Messages for OSPF’s routing database (RDB) related operations only.
9	age	Messages for OSPF’s aging process of LSAs only. LSAs are sent out on a periodic basis.
10	vlink	Messages for OSPF's virtual links operations only.
11	redist	Messages for OSPF’s route redistribution process only.
12	summary	Messages for all OSPF's summarizations only. Summarization of routes can be set for stubby areas and NSSAs.
13	dbexch	Messages for OSPF neighbors’ database exchange only.
14	hello	Messages for OSPF's hello handshaking process only.
15	auth	Messages for OSPF’s authentication process only. Authentication can be simple or MD5.
16	state	OSPF state messages only. State messages show the switch state in relation to its neighbors.
17	area	Messages for OSPF's area events only.
18	intf	Messages for OSPF’s interface operations only.
20	info	Messages for purpose to provide OSPF information only.
21	setup	Messages for OSPF’s initialization setup only.
22	time	Messages for OSPF’s time related events only. Timers are set for interfaces and LSAs.
23	mip	Messages for MIP processing of OSPF specific commands only.
24	tm	Messages for OSPF’s Task Manager communication events only.

Example of SWLOG output for RECV and SEND subapps between 2 routers:

```
-> show configuration snapshot system
! System Service:
swlog appid ospf_0 subapp 3 level debug2
swlog appid ospf_0 subapp 4 level debug2
-> show log swlog | grep ospf_0
... swlogd: ospf_0 SEND debug2(7) (2891):(1775):Sent HELLO pkt len 44, area 0.0.0.0 src
192.168.1.1 iplen 64
dst 224.0.0.5 nHop 224.0.0.5 [curTime 246533s]
... swlogd: ospf_0 RECV debug2(7) (2891):(158): HELLO pkt, intf 192.168.1.1: ipsa 192.168.1.2,
area 0.0.0.0, ip len 68
... swlogd: ospf_0 STATE info(5) :OSPF Nbr=192.168.1.2 RID=192.168.1.2 state 2WAY
... swlogd: ospf_0 SEND debug2(7) (2891):(1775):Sent HELLO pkt len 48, area 0.0.0.0 src
192.168.1.1 iplen 68
dst 224.0.0.5 nHop 224.0.0.5 [curTime 246543s]
... swlogd: ospf_0 SEND debug2(7) (2891):(1775):Sent HELLO pkt len 48, area 0.0.0.0 src
192.168.1.1 iplen 68
    dst 224.0.0.5 nHop 224.0.0.5 [curTime 246553s]
... swlogd: ospf_0 SEND debug2(7) (2891):(1775):Sent HELLO pkt len 48, area 0.0.0.0 src
192.168.1.1 iplen 68
    dst 224.0.0.5 nHop 224.0.0.5 [curTime 246563s]
... swlogd: ospf_0 RECV debug2(7) (2891):(158): DBDESC pkt, intf 192.168.1.1: ipsa 192.168.1.2,
area 0.0.0.0, ip len 52
... swlogd: ospf_0 SEND debug2(7) (2891):(1775):Sent HELLO pkt len 48, area 0.0.0.0 src
192.168.1.1 iplen 68
    dst 224.0.0.5 nHop 224.0.0.5 [curTime 246573s]
... swlogd: ospf_0 SEND debug2(7) (2891):(1775):Sent DBDESC pkt len 32, area 0.0.0.0 src
192.168.1.1 iplen 52
    dst 192.168.1.2 nHop 192.168.1.2 [curTime 246573s]
... swlogd: ospf_0 SEND debug2(7) (2891):(1775):Sent DBDESC pkt len 52, area 0.0.0.0 src
192.168.1.1 iplen 72
    dst 192.168.1.2 nHop 192.168.1.2 [curTime 246573s]
... swlogd: ospf_0 RECV debug2(7) (2891):(158): DBDESC pkt, intf 192.168.1.1: ipsa 192.168.1.2,
area 0.0.0.0, ip len 72
... swlogd: ospf_0 RECV debug2(7) (2891):(1386):[curTime=246573s] Scheduling send LSReqs
[REQ_COUNT=1]...
... swlogd: ospf_0 SEND debug2(7) (2891):(876):[curTime=246573s] (Pkt#1, #1 LSReqs), intf
192.168.1.1, dest 192.168.1.2
... swlogd: ospf_0 SEND debug2(7) (2891):(1775):Sent LSREQ pkt len 36, area 0.0.0.0 src
192.168.1.1 iplen 56 dst 192.168.1.2
    nHop 192.168.1.2 [curTime 246573s]
... swlogd: ospf_0 SEND debug2(7) (2891):(893):[curTime=246573s] Intf 192.168.1.1 dest
192.168.1.2. Sent #1 pkts, #1 LSReqs
... swlogd: ospf_0 RECV debug2(7) (2891):(158): LSREQ pkt, intf 192.168.1.1: ipsa 192.168.1.2,
area 0.0.0.0, ip len 56
... swlogd: ospf_0 SEND debug2(7) (2891):(1775):Sent LSUPDATE pkt len 64, area 0.0.0.0 src
192.168.1.1 iplen 84
    dst 192.168.1.2 nHop 192.168.1.2 [curTime 246573s
... swlogd: ospf_0 RECV debug2(7) (2891):(158): LSUPDATE pkt, intf 192.168.1.1: ipsa 192.168.1.2,
area 0.0.0.0, ip len 84
... swlogd: ospf_0 RECV debug2(7) (2891):(1542): [curTime=246573s] Rcvd #1 LSAs from Nbr
192.168.1.2, Intf 192.168.1.1
... swlogd: ospf_0 RECV debug2(7) (2891):(1565): Parsed Rcvd LS UPD msg for LSA #1
... swlogd: ospf_0 RECV debug2(7)             LSA: Type 1,lsId 192.168.1.2,advRtr 192.168.1.2,seq
0x80000002,chkSum 0x9c73,age 5
... swlogd: ospf_0 RECV debug2(7) (2891):(1712): (0.0.0.0) Nbr 192.168.1.2: New LSA
(1,192.168.1.2/192.168.1.2)
... swlogd: ospf_0 RECV debug2(7) (2891):(2473):Processing LSA from Intf 192.168.1.1, Nbr
192.168.1.2
... swlogd: ospf_0 RECV debug2(7) LSA: Type 1, LS Id 192.168.1.2, AdvRtr 192.168.1.2, Length 36,
Age 5
... swlogd: ospf_0 STATE info(5) :OSPF Nbr=192.168.1.2 RID=192.168.1.2 state FULL
```

For more refined control, one can tweak the internal log levels with a "debug" cli command. Internally, the ospf log levels are identical to the "drclog" logging levels that may be familiar to users of AOS 6 releases:

- level = 1 => info
- level = 50 => errors

- level = 60 => informative
- level = 75 => detailed
- level = 255 => all

Example:

```
debug ip ospf set lsdb 100
```

13. Troubleshooting BGP

Summary of the commands in this chapter is listed here:

```
show ip routes
show ip router database
show ip bgp routes <ip-addr> <mask>
show ip bgp path
show ip bgp path <ip-addr> <mask>
show ip bgp path neighbor-adv
show ip bpg path neighbor-rcv
show ip redist
show ip bgp network
show ip bgp neighbors
```

13.1. BGP process

Summary and comparing with CISCO

Step Cisco key commands	AOS key command
1. Load BGP & Build the neighbors	
neighbor (with-in router bgp)	ip bgp neighbor & status enable
2. Build the BGP table	
Show network (with-in router bgp)	ip bgp network & status enable
redistribute (with-in router bgp)	ip redist & status enable
3. Exchange BGP routes with neighbors	
aspathlist, prefix-list,	selectively permit
route-map	set policy – metrics, local-pref…
show ip bgp neighbor <ip-addr> advertised routes	show ip bgp path neighbor-[rcv |adv] <ip-addr>
4. Build the ip routing table	
with-in bgp: 9 step include local-pref	between routing protocols: distance / route-pref

Load BGP & Build the Neighbors

```
ip load bgp
ip bgp autonomous-system  <as>                  !64512–64534 private
ip bgp status enable                            !default is disable
ip bgp neighbor <ip addr>                       !create peer
ip bgp neighbor <ip addr> remote-as <as>        !assign remote as
ip bgp neighbor <ip addr> next-hop-self         !standard practice
            add: update-source LoopBack0        !if multiple paths
            add: ebgp-multihop                  !not directly attached
ip bgp neighbor <id addr> md5 <key>             !check logic for key
ip bgp neighbor <ip addr> status enable         !default is disable
```

Build the Neighbor - Troubleshooting Commands

```
show ip bgp                                     !global settings
show ip bgp statistics                          !global statistics
show ip bgp neighbors                           !address, AS,  state, …
CHECKS:  ip addresses, ASN, Router-ID, MD5
show ip bgp neighbor <ip-addr>                  !per neighbor details
show ip bgp neighbors statistics                          !bgp message stats
show ip bgp routes <ip-addr><mask>
show ip bgp path neighbor{-rcv | -adv} <ip-addr>
show ip bgp path ip-addr <ip-addr> <mask>                 !detail – includes MED
show ip bgp dampening [stats}
ip bgp neighbor ip_address clear soft {in | out}          !reset policies
```

Build the BGP Table – Configuration

```
[no] ip bgp network <ip addr> <mask>
ip bgp network <ip addr> <mask> metric <metric>           //exact match
ip bgp network <ip addr> <mask> admin-state enable         (a “triplet”, default: disable)
ip bgp aggregate-address <ip addr> <mask>
ip bgp aggregate-address <ip addr> <mask> summary-only     !default
ip bgp aggregate-address <ip addr> <mask> metric <value>    !optional
ip bgp aggregate-address <ip addr> <mask>  admin-state enable
 ( a “quadlet”–define, summarize, enable, default: disable)
AND/OR see ip redist command set (redist & route map)
```

Note: default route via network, redist or default originate

Route Exchange Policy Configuration

```
ip bgp policy prefix-list <name> <net add> <mask>
ip bgp policy prefix-list <name> <net add> <mask> action permit
ip bgp policy prefix-list <name> <net add> <mask> admin-state enable
(“triplet – define, action, enable, default:  admin state disable)
see also aspath-list & route-map
Use with:  ip bgp neighbor <ip.addr> in-prefixlist <name>
& ip bgp neighbor <ip.addr> out-prefixlist <name>
Check with:  show ip bgp path neighbor{-rcv | -adv} <ip-addr
MED policies : network, aggregate or policy route map
ip bgp network network_address ip_mask metric value
(CLI – 24-53, Network Configuration Guide 3-34)
ip bgp aggregate-address ip_address ip_mask [metric metric] [summary-only
ip bgp aggregate-address ip_address ip_mask [status {enable | disable}]
```

BGP Metric with a Route Map & Assign to Neighbor

```
ip bgp policy route-map <m1> 90 prefix-list med-filter          !name & seq
ip bgp policy route-map <m1> 90 action permit                    !action
             ADD EXAMPLES  prefix-list | many others            !conditions
ip bgp policy route-map <m1> 90 med 10                          !attrib actions
ip bgp policy route-map <m1> 90 med-mode rep
                                                                !method --the default is none (not helpful)
ip bgp policy route-map <m1> 90 admin-state enable   !enable
ip bgp neighbor <ip_addr> route-map <m1> filterout |in !check
or                  <ip_addr> route-map <m1> filter-in
ip bgp neighbor clear soft [in | out]                            !re-config
show ip route-map
```

Border Router Redistribution Policies

references a route map

```
ip redist local into ospf route-map "allow" admin-state enable       note:  default is admin-state enable
ip redist local into bgp route-map "allow" admin-state enable
ip redist static into ospf route-map "allow" admin-state enable
ip redist static into bgp route-map "allow" admin-state enable
ip redist bgp into ospf route-map "allow" admin-state enable
```

Note: ip bgp network statements are an option that supports MED
Note: policy-lists provide an option to limits the advertised prefixes.
Need an IPv6 method to always advertise the /48 – Cisco null zero address equivalent for a static route with-in a block, 4-53

Build the BGP Table - Troubleshooting Commands

```
show ip routes                                  !current routing table
show ip router database                         !all routes/all protocols
show ip bgp routes [<ip-addr> <mask>            !BGP routes
show ip bgp path                                !next hop & BGP attr for paths
show ip bgp path <ip-addr> <mask>               !adds path detail  pref, MED..
show ip bgp path neighbor-adv                   !show advertised routes to neighbor
        Similar to Cisco show ip bpg neighbor <ip-addr advertised-routes
show ip bpg path neighbor-rcv                   !1st source – BGP
 Similar to Cisco show ip brp neighbor <ip-addr> routes
show ip redist                                      !2nd source – redist
show ip bgp network                                 !3rd soure – network
```

Build BGP Route Table Route Selection:

Next Hop Reachable
Weight (Cisco proprietary)
Highest bgp local preference (default 100)
Locally injected routes (locally injected is better than i/eBGP)
Fewest autonomous systems in AS Path
AS path origin (IGP < EGP < Incomplete)
Lowest Multi-Exit Discriminator (MED)
Neighbor type (EBGP before IBGP)
IGP metric to next hop (closer is better)
Source of route (IGP <EBGP <IBGP)
Lower BGP Router ID
Oldest eBGP < smallest Neighbor RID < smallest neighbor IP

Build the Local Forwarding Table

```
show ip routes                              //forwarding table & source
show ip router database                     //inspect the available paths
show ip route protocol [bgp | ospf…]        //aka Cisco sh ip bgp
default route preference                    //select the routes based on lowest
```

IPv4 route preference (adminstrative dist)

static	2	ospf	110
isis	118	isisl2	115
rip	120	ebgp	190
ibgp	200	import	210

```
ip route-pref {static |ospf | ibgp | ebgp | import}  <value>
```

BGP - Syslog Tools

```
swlog appid bgp_0 subapp all enable
swlog appid bgp_0 subapp all level debug3
swlog log swlog | grep bgp | tail
```

Note: If using multiple vrf, log by vrf name. Replace bgp_0 with bgp_<vrf_name> .

ROUTING Troubleshooting Flow

Other issues with “routing” symptoms:
MTU mismatch, unidirectional link, duplex mismatch, link errors, L2 config errors, QOS rules, TTL setting, mismatched subnet masks, ip helper, ip-source-filtering, port security?
Then
Redistribution configuration
Protocols not advertising a route when “intended”
Routes not redistributed when “intended”
Incorrect route filtering with prefix list / mask
“Start in the middle” approach

```
show ip protocols               //which protocols are loaded
show ip [bgp | ospf |…          //admin up, operational up
show ip route summary           //adding any routes to routing table?
show ip route-pref              //are “ad’s” at default?
show ip redist
show ip route-map
show ip access-list
show ip bgp policy prefix-list
show ip router database
show ip route [gateway | protocol | summary | destination]
show ip interface [<int name>]
```

BGP v6 additions

```
ipv6 bgp unicast
ipv6 redist static into bgp route-map static

ipv6 bgp network 2001:470:4979:1::/64
ipv6 bgp network 2001:470:4979:1::/64 status enable

ipv6 interface "v6if-v99" vlan 99
ipv6 address 2001:470:4979:99::1/64 "v6if-v99"

ipv6 bgp neighbor 2001:470:4979:99::24
ipv6 bgp neighbor 2001:470:4979:99::24 timers 10 30
ipv6 bgp neighbor 2001:470:4979:99::24 remote-as 2152
ipv6 bgp neighbor 2001:470:4979:99::24 activate-ipv6
ipv6 bgp neighbor 2001:470:4979:99::24 status enable
```

13.2. **Advance Troubleshooting**

```
-> show ip bgp neighbors
Legends: Nbr = Neighbor
         As  = Autonomous System
Nbr address     As          Admin state Oper state   BGP Id          Up/Down     BFD Status
---------------+-----------+-----------+------------+---------------+-----------+----------
192.168.10.1    65555       enabled     established  192.168.10.1    00h:37m:46s disabled

-> show ip bgp neighbors 192.168.10.1
Neighbor address                  = 192.168.10.1,
Neighbor autonomous system        = 65555,
Neighbor Admin state              = enabled,
Neighbor Oper state               = established,
Neighbor passive status           = disabled,
Neighbor name                     = peer(192.168.10.1),
Neighbor local address            = vlan10,
Neighbor EBGP multiHop            = disabled,
Neighbor next hop self            = disabled,
Neighbor Route Refresh            = enabled,
Neighbor Ipv4 unicast             = enabled,
Neighbor Ipv4 multicast           = disabled,
Neighbor type                     = internal,
Neighbor auto-restart             = enabled,
Neighbor route-reflector-client   = disabled,
Neighbor confederation status     = disabled,
Neighbor remove private AS        = disabled,
Neighbor default originate        = disabled,
Neighbor maximum prefixes         = 5000,
Neighbor max prefixes warning     = enabled,
# of prefixes received            = 0,
Neighbor MD5 key                  = <none>,
Neighbor local port               = 179,
Neighbor TCP window size          = 32768,
Graceful Restart State            = NotRestarting,
Advertised Restart Interval       = 90s,
Forwarding State during restart   = NotPreserved,
Activate IPv6 unicast             = disabled,
Configured IPv6 NextHop Address   = ::,
Neighbor IPv6 unicast             = not-advertised,
BFD Status                        = Disabled
-> show ip bgp
aggregate-address  dampening-stats    network            policy             statistics
dampening          neighbors          path               routes
-> show ip bgp statistics
# of Active Prefixes Known                = 0,
# of EBGP Neighbors in Established State  = 0,
# of IBGP Neighbors in Established State  = 1,
# of Feasible Paths                       = 0,
# of Dampened Paths                       = 0,
# of Unsynchronized Paths                 = 0,
# of Policy unfeasible paths              = 0,
Total Number of Paths                     = 0
```

All DRC logs were moved to SWLOG. Logging is avaiable per VRF (please replace "0" in the example below with your VRF name).

```
-> swlog appid bgp_0 subapp all enable
-> swlog appid bgp_0 subapp all level debug3
-> show log swlog | grep bgp | tail
Nov 13 22:27:20 (none) swlogd: bgp_0 ka debug1(6) vrfId 0: [peer(192.168.10.1),65555] <=
KeepAlive msg
Nov 13 22:27:20 (none) swlogd: bgp_0 fsm debug2(7) [peer(192.168.10.1),65555] restarting hold
timer [90] sec
Nov 13 22:27:20 (none) swlogd: bgp_0 fsm debug1(6) vrfId 0: [peer(192.168.10.1),65555] EXIT:
EVENT keepalive_msg_recv_event [NEXT_STATE Established]
Nov 13 22:27:23 (none) swlogd: bgp_0 info debug3(8) [peer(192.168.10.1),65555]  START at Time
02h:42m:08s:000ms
```

```
Nov 13 22:27:23 (none) swlogd: bgp_0 info debug3(8) [peer(192.168.10.1),65555]  END at Time
02h:42m:08s:000ms
Nov 13 22:27:35 (none) swlogd: bgp_0 peer debug2(7) bgpTask 0x101600f8,bgp_env 0x101678e8,peer
0x1016d548
Nov 13 22:27:35 (none) swlogd: bgp_0 updtx debug2(7) vrfId 0: [peer(192.168.10.1),65555] RouteAdv
timer expired
Nov 13 22:27:35 (none) swlogd: bgp_0 peer debug3(8) vrfId 0: [peer(192.168.10.1),65555] Added
route advrtsmt timer for 26s
Nov 13 22:27:35 (none) swlogd: bgp_0 info debug3(8) [peer(192.168.10.1),65555]  START at Time
02h:42m:20s:000ms
Nov 13 22:27:35 (none) swlogd: bgp_0 info debug3(8) [peer(192.168.10.1),65555]  END at Time
02h:42m:20s:000ms
```

14. Troubleshooting IP Multicast Switching (IPMS)

- There should be 1 dedicated querier in the network (in case more than 1 querier is enabled, then there is 1 querier elected and the others remain inactive)
- Querier-forwarding should be enabled only on switches located between multicast sources and the querier
- Zapping should be enabled only on edge devices
- Proxying may be enabled on all devices
- IGMP messages must be sent with TTL equal 1
- Freezing and pixelation might be caused by congestion or incorrect IGMP Leave handling

Summary of the commands in this chapter is listed here:

```
show ip multicast
show ip multicast querier
show ip multicast source
show ip multicast group
show ip multicast group
show log swlog | grep -E "ipmsCmm|ipmsNi"
debug ip multicast member
debug ip multicast flow
debug ip multicast channel
debug ip multicast interface
debug ip multicast vlan xx
debug ip multicast stats
debug $(pidof ipmscmm) 'p ipms::bcm_max'
getreg MC_CONTROL_5
d chg l2mc
d l3_ipmc
d l3_entry_2
d l3_ipmc
```

14.1. Introduction

OS6900 and OS10K Hardware Limitations

ASICs Platforms	Ingress Limits		Egress Limits
OS10K	**L3_IPV4_MULTICAST**	**L3_IPV6_MULTICAST**	**L3_IPMC**
OS10K-GNI-C48E			
OS10K-GNI-U48E	8K	4K	4K
OS10K-XNI-U32S			
OS10K-XNI-U32S	8K	4K	2K
OS6900	4K	2K	4K

OS6860 Hardware Limitations

	Ingress limits		Egress limits
	L3_IPV4_MULTICAST (L3_ENTRY_2)	**L3_IPV6_MULTICAST (L3_ENTRY_2)**	**L3_IPMC**
OS6860	12K	6K	8K

On the ingress side, a single lookup hash table (L3_ENTRY) is used for classifying IPv4 and IPv6 IPMC flows in addition to storing the IPv4 and IPv6 unicast host entry caches. For example, a single BCM56340 ASIC could only support the maximum number of IPv6 IPMC forwarding entries if there were no IPv4 or IPv6 host entries or IPv4 IPMC forwarding entries, which will generally never be the case. Additionally, hash collisions within the table may prevent the maximum number of entries from being installed in the table.
On the egress side, a single lookup table (L3_IPMC) provides IPMC replication resources for switching/routing IPv4 and IPv6 IPMC traffic and for Ethernet services like VPLS and PBB.
By aggregating traffic across multiple ASICs, it becomes possible to leverage the ingress classification abilities of each ASIC to support a larger number of flows than could otherwise be possible with just a single ASIC. However, the point-to-multipoint nature of IPMC forwarding requires that egress resources be synchronized in order to support distribution across ASICs. In a Virtual Chassis, the maximum number of ingress lookup entries for the overall system becomes the sum of the underlying ASICs while the number of IPMC egress resources remains fixed to the lowest common denominator amongst the ASICs.

OS6900 and OS10K Hardware Limitations

Shared L3_ENTRY table

Hardware entries are shared between IPv4 host table, IPv6 host table, IPv4 multicast and IPv6 multicast. An ARP entry in the IPv4 host table consumes 1 hash entry. An ND in the IPv6 host table consumes 2 entries. An IPv4 multicast entry consumes 2 entries and an IPv6 multicast entry consumes 4 entries. For example we can learn up to 8K IPv4 hosts in the ARP table only if there are no multicast flows classified. The maximum number or multicast flows is controlled by the MC_CONTROL_5 register using arguments SHARED_TABLE_L2MC_SIZE and SHARED_TABLE_IPMC_SIZE.

IPMC forwarding index range

Restrict the IPMC forwarding index range to just the reserved indexes:

```
capability ipmc-max-entry 3
```

Enabling IPMS

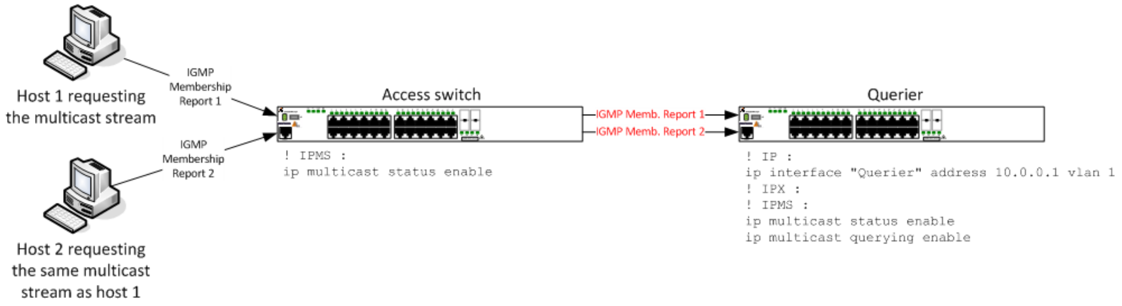
When IPMS is enabled all IP packets matching destination IP address 224.0.0.0/24 are copied to CPU. These packets are rate limited to avoid CPU flooding. An IGMP Membership Query is broadcasted in all VLANs just after IPMS is enabled. Example (generated by a switch without any IP interface configured):

```
MAC: ------  MAC Header  ------
MAC:
MAC: Destination Address : 01 00 5E 00 00 01
MAC: Source Address      : 00 E0 B1 A6 C0 9C
MAC: Type                : 0x0800 (Ethernet II)
MAC:
IP: ------  IP Header  -----------
IP:
IP: Version                        = 04 (0x04)
IP: Header Length                  = 24 (0x18)
IP: Differentiated Services Field  = 192 (0xC0)
IP: Type of Service                = 192 (0xC0)
IP: Total Length                   = 32 (0x0020)
IP: Identification                 = 4229 (0x1085)
IP: Fragment Offset                = 0
IP: Time to Live                   = 1 (0x01)
IP: Protocol                       = IGMP
IP: Checksum                       = 0x…
```

```
IP: Source Address                  = 0.0.0.0
IP: Destination Address             = 224.0.0.1
IP: Options & Padding               = 0x94040000
IP:
IGMP: ------  IGMP Header -----------
IGMP:
IGMP: Version     =  2  (0x2)
IGMP: Type        = Membership Query (0x11)
IGMP: Max Response Time  =  25  (0x19)
IGMP: Checksum    =  0xEEE6
IGMP: Group       =  0.0.0.0
IGMP:
```

Proxying

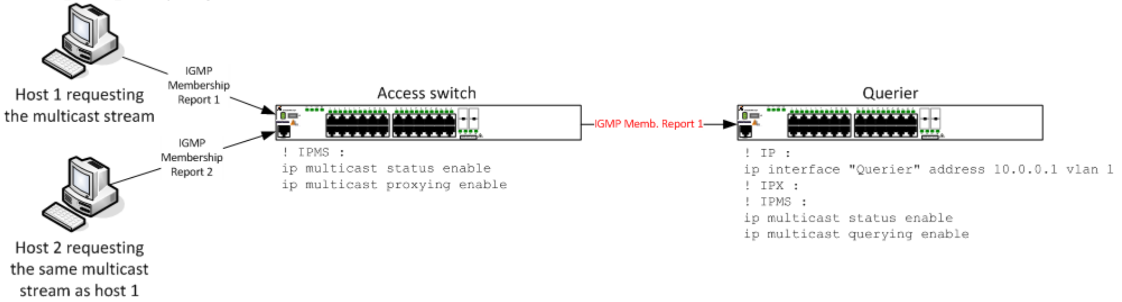
The proxying feature is used to optimize IGMP Membership Report traffic between an access switch and a querier. The access switch creates a table of registered subscribers containing IP address of multicast group, a port, VLAN id, IGMP version, mode and SSM address. Each time a new IGMP Membership Report is received, the access switch verifies if it is a new registration (new IP multicast group in a VLAN). If necessary (if it is a new registration) IGMP Membership Report is forwarded to the querier. If the group in IGMP Membership Report is already registered there's no need to forward any messages to the querier (there's no need to update querier's registration database), only the table of registered subscribers on the access switch is updated with the port on which the new registration was received. Example (proxying disabled):



```
! IPMS :
ip multicast status enable
```

```
! IP :
ip interface "Querier" address 10.0.0.1 vlan 1
! IPX :
! IPMS :
ip multicast status enable
ip multicast querying enable
```

Example (proxying enabled):


Host 1 requesting the multicast stream
IGMP Membership Report 1
IGMP Membership Report 2
Host 2 requesting the same multicast stream as host 1
Access switch
IGMP Memb. Report 1
Querier

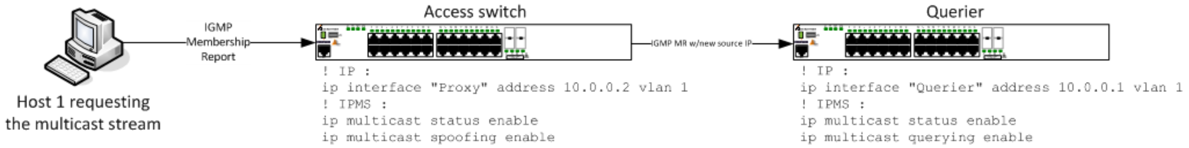
```
! IPMS :
ip multicast status enable
ip multicast proxying enable
```

```
! IP :
ip interface "Querier" address 10.0.0.1 vlan 1
! IPX :
! IPMS :
ip multicast status enable
ip multicast querying enable
```

Note: Proxying enabled case. All IGMP Membership Reports generated by the access switch have IP source address inherited from the first received IGMP Membership Reports for a specific IP multicast group.

Spoofing

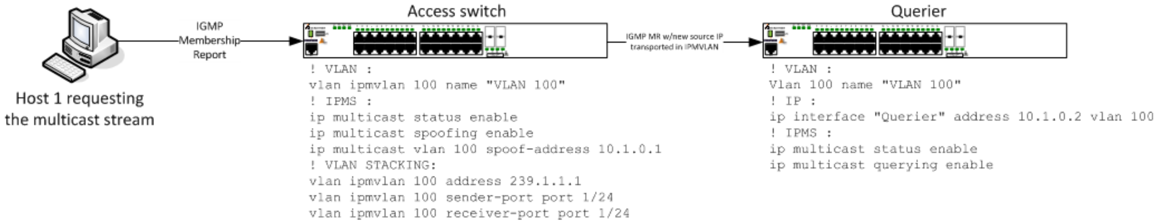
When the spoofing feature is enabled, source IP address of each IGMP Membership Report is updated according to IP interface address in the VLAN in which the IGMP Membership Report was received. Other IGMP messages are not affected. Example (without IPMVLAN):



```
! IP :
ip interface "Proxy" address 10.0.0.2 vlan 1
! IPMS :
ip multicast status enable
ip multicast spoofing enable
```

```
! IP :
ip interface "Querier" address 10.0.0.1 vlan 1
! IPMS :
ip multicast status enable
ip multicast querying enable
```

Example (with IPMVLAN):



```
! VLAN :
vlan ipmvlan 100 name "VLAN 100"
! IPMS :
ip multicast status enable
ip multicast spoofing enable
ip multicast vlan 100 spoof-address 10.1.0.1
! VLAN STACKING:
vlan ipmvlan 100 address 239.1.1.1
vlan ipmvlan 100 sender-port port 1/24
vlan ipmvlan 100 receiver-port port 1/24
```

```
! VLAN :
Vlan 100 name "VLAN 100"
! IP :
ip interface "Querier" address 10.1.0.2 vlan 100
! IPMS :
ip multicast status enable
ip multicast querying enable
```

Note:

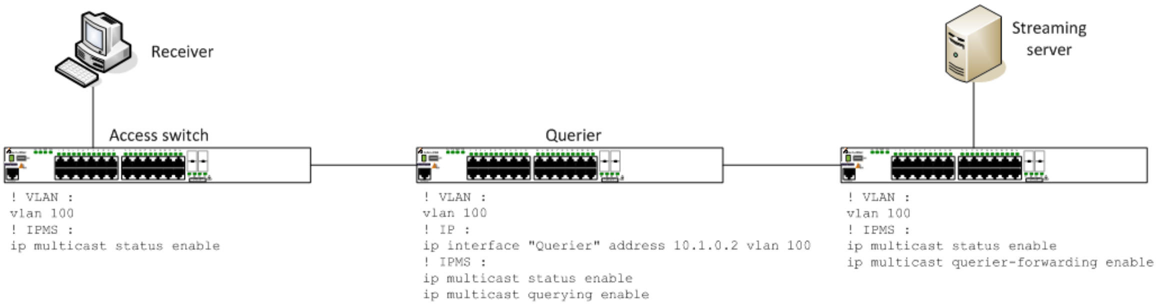
- The querier is not able to differentiate IGMP Membership Reports based on IP source address basis when spoofing feature is enabled.
- This feature should be used along with IPMVLAN feature to ensure that IGMP Membership Reports are forwarded with a source IP address belonging to an IP network range in the VLAN configured on the querier switch, which is corresponding to the IPMVLAN.
- The querier switch in AOS implementation accepts IGMP Membership Reports even if their source IP address is not matching IP address network in this VLAN.

Zapping

If zapping feature is enabled multicast groups are immediately removed from registered subscribers table after receiving IGMP Leave (IGMP version 2) message or IGMP Membership Report type 4 (IGMP version 3).

Querier-forwarding

Querier-forwarding feature should be enabled if a streaming device is connected to a switch, which is not a querier. If this feature is enabled on a switch, then all multicast streams are forwarded to the querier.



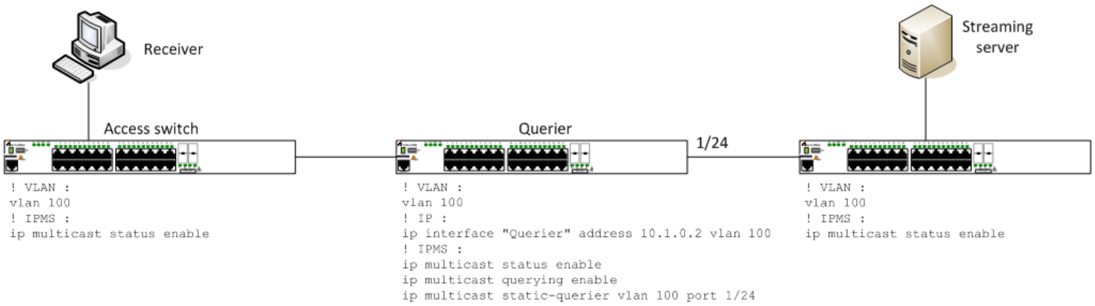
```
! VLAN :
vlan 100
! IPMS :
ip multicast status enable
```

```
! VLAN :
vlan 100
! IP :
ip interface "Querier" address 10.1.0.2 vlan 100
! IPMS :
ip multicast status enable
ip multicast querying enable
```

```
! VLAN :
vlan 100
! IPMS :
ip multicast status enable
ip multicast querier-forwarding enable
```

Static-querier

Static-querier feature should be enabled if a streaming device is connected to a switch, which is not a querier. If this feature is enabled on a querier, then all IGMP Membership queries are forwarded towards the streaming device.



```
! VLAN :
vlan 100
! IPMS :
ip multicast status enable
```

```
! VLAN :
vlan 100
! IP :
ip interface "Querier" address 10.1.0.2 vlan 100
! IPMS :
ip multicast status enable
ip multicast querying enable
ip multicast static-querier vlan 100 port 1/24
```

```
! VLAN :
vlan 100
! IPMS :
ip multicast status enable
```

Neighbor

The "neighbor" concept was introduced for IGMPv3 in RFC 3376. This concept is also used for IGMPv2. The extract from the RFC 3376:

6. Description of the Protocol for Multicast Routers The information gathered by IGMP is provided to whichever multicast routing protocol is being used by the router, in order to ensure that multicast packets are delivered to all networks where there are interested receivers.
In AOS each multicast router becomes a "dynamic neighbor" by default. AOS switches replicate all multicast packets and all IGMP messages to all neighbors due to compliance with RFC 3376.

IPMVLAN scenarios in Ethernet-Services mode

Not supported in AOS 7 and AOS 8

Example of IGMP version 2 messages

IGMP Membership Report

Note: All IGMP messages need to have TTL field set to 1. Messages with other values are dropped.
In IGMP version 2 a group address which is registered by IGMP Membership Report corresponds to a destination IP address of a packet.

```
MAC: ------  MAC Header  ------
MAC:
MAC: Destination Address : 01 00 5E 01 01 01
MAC: Source Address      : 00 00 00 08 73 F0
MAC: Type                : 0x0800 (Ethernet II)
MAC:
IP: ------  IP Header  -----------
IP:
IP: Version                        = 04 (0x04)
IP: Header Length                  = 20 (0x14)
IP: Differentiated Services Field  = 0 (0x00)
IP: Type of Service                = 00 (0x00)
IP: Total Length                   = 28 (0x001C)
IP: Identification                 = 0 (0x0000)
IP: Fragment Offset                = 0
IP: Time to Live                   = 1 (0x01)
IP: Protocol                       = IGMP
IP: Checksum                       = 0x…
IP: Source Address                 = 10.0.0.111
IP: Destination Address            = 239.1.1.1
IP:
IGMP: ------  IGMP Header -----------
IGMP:
IGMP: Version    =  2  (0x2)
IGMP: Type       = Membership Report v2 (0x16)
IGMP: Max Response Time  =  100  (0x64)
IGMP: Checksum   =  0xF796
IGMP: Group      =  239.1.1.1
IGMP:
```

IGMP Leave

Note: In IGMP version 2 a leave message is always sent with 224.0.0.2 IP destination address.

```
MAC: ------  MAC Header  ------
MAC:
MAC: Destination Address : 01 00 5E 00 00 02
MAC: Source Address      : 00 00 00 08 73 F0
MAC: Type                : 0x0800 (Ethernet II)
MAC:
IP: ------  IP Header  -----------
IP:
IP: Version                        = 04 (0x04)
IP: Header Length                  = 20 (0x14)
IP: Differentiated Services Field  = 0 (0x00)
IP: Type of Service                = 00 (0x00)
IP: Total Length                   = 28 (0x001C)
IP: Identification                 = 0 (0x0000)
IP: Fragment Offset                = 0
```

```
IP: Time to Live                   = 1 (0x01)
IP: Protocol                       = IGMP
IP: Checksum                       = 0x…
IP: Source Address                 = 10.0.0.111
IP: Destination Address            = 224.0.0.2
IP:
IGMP: ------  IGMP Header -----------
IGMP:
IGMP: Version    =  2  (0x2)
IGMP: Type       = Leave Group (0x17)
IGMP: Max Response Time  =  100  (0x64)
IGMP: Checksum   =  0xF68C
IGMP: Group      =  239.1.1.1
```

IGMP Membership Query

Note:

- In IGMP version 2 a Membership Query message is always sent with 224.0.0.1 IP destination address.
- An IGMP Membership Query message is used to verify if there are still active subscribers registered for a specific group.
- IGMP Membership Query with 0.0.0.0 multicast group is called a General Query message. It used for announcing querier's availability.

```
MAC: ------  MAC Header  ------
MAC:
MAC: Destination Address : 01 00 5E 00 00 01
MAC: Source Address      : 00 00 00 00 0C 00
MAC:
IP: ------  IP Header  -----------
IP:
IP: Version                        = 04 (0x04)
IP: Header Length                  = 20 (0x14)
IP: Differentiated Services Field  = 0 (0x00)
IP: Type of Service                = 00 (0x00)
IP: Total Length                   = 28 (0x001C)
IP: Identification                 = 0 (0x0000)
IP: Fragment Offset                = 0
IP: Time to Live                   = 1 (0x01)
IP: Protocol                       = IGMP
IP: Checksum                       = 0x…
IP: Source Address                 = 10.0.0.1
IP: Destination Address            = 224.0.0.1
IP:
IGMP: ------  IGMP Header -----------
IGMP:
IGMP: Version    =  2  (0x2)
IGMP: Type       = Membership Query (0x11)
IGMP: Max Response Time  =  50  (0x32)
IGMP: Checksum   =  0xEECD
IGMP: Group      =  0.0.0.0
IGMP:
```

Example of IGMP version 3 messages

IGMP Membership Report type 3

Note: IGMP Membership Report type 3 is used for registration of new subscribers. One message can be used to register multiple multicast groups.
IGMP Membership Report in IGMP version 3 is always sent with 224.0.0.22 destination IP address.

```
MAC: ------  MAC Header  ------
MAC:
```

```
MAC: Destination Address : 01 00 5E 00 00 16
MAC: Source Address      : 00 00 00 08 73 F1
MAC:
IP: ------  IP Header  -----------
IP:
IP: Version                        = 04 (0x04)
IP: Header Length                  = 20 (0x14)
IP: Differentiated Services Field  = 0 (0x00)
IP: Type of Service                = 00 (0x00)
IP: Total Length                   = 36 (0x0024)
IP: Identification                 = 0 (0x0000)
IP: Fragment Offset                = 0
IP: Time to Live                   = 1 (0x01)
IP: Protocol                       = IGMP
IP: Checksum                       = 0x…
IP: Source Address                 = 10.0.0.111
IP: Destination Address            = 224.0.0.22
IP:
IGMP: ------  IGMP Header -----------
IGMP:
IGMP: Version    =  3  (0x3)
IGMP: Type       = Membership Report v3 (0x22)
IGMP: Reserved   =  0  (0x0)
IGMP: Checksum   =  0xE8F0
IGMP: Reserved   =  0  (0x0)
IGMP: Number of Group Records =   1  (0x1)
IGMP:      ------  IGMP Group Record -----------
IGMP:
IGMP: Record Type         =  Change To Include Mode  (0x3)
IGMP: Aux Data Len        =  0  (0x0)
IGMP: Number of Source(s) = 0
IGMP: Multicast Address   = 239.1.1.1
IGMP:
```

IGMP Membership Report type 4

Note: IGMP Membership Report type 4 is used for unregistering subscribers. One message can be used to unregister multiple multicast groups.
IGMP Membership Report in IGMP version 3 is always sent with 224.0.0.22 destination IP address.

```
MAC: ------  MAC Header  ------
MAC:
MAC: Destination Address : 01 00 5E 00 00 16
MAC: Source Address      : 00 00 00 08 73 F1
MAC:
IP: ------  IP Header  -----------
IP:
IP: Version                        = 04 (0x04)
IP: Header Length                  = 20 (0x14)
IP: Differentiated Services Field  = 0 (0x00)
IP: Type of Service                = 00 (0x00)
IP: Total Length                   = 36 (0x0024)
IP: Identification                 = 0 (0x0000)
IP: Fragment Offset                = 0
IP: Time to Live                   = 1 (0x01)
IP: Protocol                       = IGMP
IP: Checksum                       = 0x1708
IP: Source Address                 = 10.0.0.111
IP: Destination Address            = 224.0.0.22
IP:
IGMP: ------  IGMP Header -----------
IGMP:
IGMP: Version    =  3  (0x3)
IGMP: Type       = Membership Report v3 (0x22)
IGMP: Reserved   =  0  (0x0)
IGMP: Checksum   =  0xE7F0
IGMP: Reserved   =  0  (0x0)
IGMP: Number of Group Records =   1  (0x1)
IGMP:      ------  IGMP Group Record -----------
```

```
IGMP:
IGMP: Record Type         =  Change To Exclude Mode  (0x4)
IGMP: Aux Data Len        =  0  (0x0)
IGMP: Number of Source(s) = 0
IGMP: Multicast Address   = 239.1.1.1
IGMP:
```

IGMP Membership Query

Note: An IGMP Membership Query message is used to verify if there are still active subscribers registered for a specific group.

IGMP Membership Query with 0.0.0.0 multicast group is called a General Query message. It used for announcing querier's availability.

```
MAC: ------  MAC Header  ------
MAC:
MAC: Destination Address : 01 00 5E 00 00 01
MAC: Source Address      : 00 00 00 00 0C 00
MAC:
IP: ------  IP Header  -----------
IP:
IP: Version                        = 04 (0x04)
IP: Header Length                  = 20 (0x14)
IP: Differentiated Services Field  = 0 (0x00)
IP: Type of Service                = 00 (0x00)
IP: Total Length                   = 32 (0x0020)
IP: Identification                 = 0 (0x0000)
IP: Fragment Offset                = 0
IP: Time to Live                   = 1 (0x01)
IP: Protocol                       = IGMP
IP: Checksum                       = 0x...
IP: Source Address                 = 10.0.0.1
IP: Destination Address            = 224.0.0.1
IP:
IGMP: ------  IGMP Header -----------
IGMP:
IGMP: Version    =  3  (0x3)
IGMP: Type       = Membership Query (0x11)
IGMP: Max Response Time = 0x32 (Raw = 50)
IGMP: Checksum   =  0xEECD
IGMP: Group      =  0.0.0.0
IGMP: Reserved (Resv) = 0  (0x0)
IGMP: Suppress Router-Side Processing (S) = 0  (0x0)
IGMP: Querier's Robustness Variable (QRV)= 0  (0x0)
IGMP: Querier's Query Interval Code(QQIC) = 0x0 (Raw = 0)
IGMP: Number of Sources = 0  (0x0)
```

14.2. **Basic troubleshooting**

Basic outputs

Display IPMS config summary:

```
-> show ip multicast
Status                                          = enabled,
Querying                                        = disabled,
Proxying                                        = disabled,
Spoofing                                        = disabled,
Zapping                                         = disabled,
Querier Forwarding                              = disabled,
Flood Unknown                                   = disabled,
Version                                         = 2,
Robustness                                      = 2,
Query Interval (seconds)                        = 125,
Query Response Interval (tenths of seconds)     = 100,
Last Member Query Interval (tenths of seconds)  = 10,
Unsolicited Report Interval (seconds)           = 1,
Router Timeout (seconds)                        = 90,
Source Timeout (seconds)                        = 30,
Max-group                                       = 0,
Max-group action                                = none,
Helper-address                                  = 0.0.0.0,
Zero-based Query                                = enabled
```

Display querier information:

```
-> show ip multicast querier
Total 1 Queriers
Host Address    VLAN  Port      Static  Count  Life
---------------+-----+---------+-------+------+-----
172.13.0.1       2107 1/1/1     no      28520  254
```

Display sources information:

```
-> show ip multicast source
Total 1 Sources
Group Address   Host Address    Tunnel Address  VLAN  Port
---------------+---------------+---------------+-----+---------
239.0.0.1       10.0.0.1        0.0.0.0         1     2/1/1
```

Display information related to receives:

```
-> show ip multicast group
Total 1 Groups
Group Address   Source Address  VLAN  Port      Mode     Static  Count  Life
---------------+---------------+-----+---------+--------+-------+------+-----
239.0.0.1       0.0.0.0          1    1/1/48    exclude  no      5      259
```

Display active flow information:

Warning: Flow in this table are populated only in case there is a group match from the the group table and the source table, there must be also an active querier in the network.

```
-> show ip multicast forward
Total 1 Forwards
                                                  Ingress         Egress
Group Address   Host Address    Tunnel Address  VLAN  Port      VLAN  Port
---------------+---------------+---------------+-----+---------+-----+---------
239.0.0.1       10.0.0.1        0.0.0.0         1     2/1/1     1     1/1/48
```

Logging to SWLOG

Enabling detailed logging:

```
-> show configuration snapshot system
! System Service:
swlog appid ipmsCmm subapp all level debug3
swlog appid ipmsNi subapp all level debug3
```

An example output of a single IGMPv2 Membeship Request (IGMP receiver is located is connected to port 1/1/48 and the sender to port 2/1/1):

```
-> show log swlog | grep -E "ipmsCmm|ipmsNi"
<snap> swlogd: ipmsCmm msg debug1(6) ip/6 recv len 164
<snap> swlogd: ipmsNi cap debug1(6) pd_recv/4  src 00-00-00-00-00-01 vlan 1 stack 0 modid 0 port
48 vpn 47 vp 0 cpu 1 flood 1 pd_client 0
<snap> swlogd: ipmsNi cap debug1(6) igmp type x16 vlan 1 stack 0 port 1/1/48 group 239.0.0.1 host
192.168.1.1 sa 00-00-00-00-00-01
<snap> swlogd: ipmsNi msg debug1(6) mcm report vlan 1 stack 0 port 1/1/48 vp 0 host 192.168.1.1
sa 00-00-00-00-00-01 modid 0 devport 48
<snap> swlogd: ipmsCmm msg debug1(6) cni recv len 76
<snap> swlogd: ipmsCmm msg debug1(6) cni report vlan 1 stack 0 port 1/1/48 svp 0 host 192.168.1.1
sa 00-00-00-00-00-01 modid 0 devport 48
```

```
<snap> swlogd: ipmsCmm call debug1(6) report vlan 1 stack 0 ifindex 1/1/48 host 192.168.1.1 sa
00-00-00-00-00-01 modid 0 devport 48
<snap> swlogd: ipmsCmm sub debug1(6) translate vlan 1 stack 0 ifindex 1048(1/1/48) group
239.0.0.1
<snap> swlogd: ipmsCmm sub debug1(6) policy/4 vlan 1 ifindex 1/1/48 group 239.0.0.1 host
192.168.1.1 sa 00-00-00-00-00-01
<snap> swlogd: ipmsCmm rpt debug1(6) igmp/2 join vlan 1 ifindex 1/1/48 host 192.168.1.1 group
239.0.0.1
<snap> swlogd: ipmsCmm obj debug1(6) channel add vlan 1 group 239.0.0.1
<snap> swlogd: ipmsCmm sub debug1(6) is_max_grp/4 vlan 1 ifindex 1/1/48
<snap> swlogd: ipmsCmm obj debug1(6) member add vlan 1 ifindex 1/1/48 group 239.0.0.1
<snap> swlogd: ipmsCmm call debug1(6) havlan check vlan 1 group 239.0.0.1 ifindex 1/1/48 allow 1
<snap> swlogd: ipmsCmm obj debug1(6) thread add vlan 1 group 239.0.0.1 host 10.0.0.1 next 1
<snap> swlogd: ipmsCmm rpt debug1(6) gmi timer vlan 1 ifindex 1/1/48 group 239.0.0.1
<snap> swlogd: ipmsCmm sub debug1(6) link vlan 1 group 239.0.0.1 host 10.0.0.1 next 1 port 1/1/48
<snap> swlogd: ipmsCmm obj debug1(6) fabric add vlan 1 group 239.0.0.1 host 10.0.0.1 next 1
ifindex 1/1/48
<snap> swlogd: ipmsCmm rpt debug1(6) gmi timer vlan 1 group 239.0.0.1
<snap> swlogd: ipmsCmm call debug1(6) relay ia4_t vlan 1 len 28
<snap> swlogd: ipmsCmm msg debug1(6) sec proxy msg_report4 len 68 rem_chas 2
<snap> swlogd: ipmsCmm sub debug1(6) settle fwdvecs 0 v4flows 1 v6flows 0
<snap> swlogd: ipmsCmm sub debug1(6) alloc
<snap> swlogd: ipmsCmm obj debug1(6) mcindex add index 3
<snap> swlogd: ipmsCmm obj debug1(6) seq add id 10 cookie 3
<snap> swlogd: ipmsCmm obj debug1(6) fwdvec add mcindex 3 vlan 1 ifindex 2/1/1 fwds 1 trap 0
<snap> swlogd: ipmsCmm msg debug1(6) nic collect chas 1 slot 1 index 3 enable 1
<snap> swlogd: ipmsCmm msg debug1(6) nic collect chas 2 slot 1 index 3 enable 1
<snap> swlogd: ipmsCmm msg debug1(6) nic rep chas 1 slot 1 index 3 type 1
<snap> swlogd: ipmsCmm msg debug1(6) nic rep chas 2 slot 1 index 3 type 1
<snap> swlogd: ipmsCmm msg debug1(6) nic set chas 1 slot 1 index 3 port 2/1/1
<snap> swlogd: ipmsCmm msg debug1(6) nic set chas 2 slot 1 index 3 port 2/1/1
<snap> swlogd: ipmsCmm msg debug1(6) nic mtu chas 1 slot 1 index 3 mtu 1500
<snap> swlogd: ipmsCmm msg debug1(6) nic mtu chas 2 slot 1 index 3 mtu 1500
<snap> swlogd: ipmsCmm msg debug1(6) nic up chas 1 slot 1 index 3 port 1/1/48 vlan 1 nalv 1
<snap> swlogd: ipmsCmm msg debug1(6) nic trap chas 1 slot 1 index 3 enable 0
<snap> swlogd: ipmsCmm msg debug1(6) nic trap chas 2 slot 1 index 3 enable 0
<snap> swlogd: ipmsCmm msg debug1(6) nic valid chas 1 slot 1 index 3
<snap> swlogd: ipmsCmm msg debug1(6) nic valid chas 2 slot 1 index 3
<snap> swlogd: ipmsCmm msg debug1(6) nic collect chas 1 slot 1 index 3 enable 0
<snap> swlogd: ipmsCmm msg debug1(6) nic collect chas 2 slot 1 index 3 enable 0
<snap> swlogd: ipmsCmm msg debug1(6) nic ack chas 1 slot 1 seq 10 cookie 3
<snap> swlogd: ipmsCmm msg debug1(6) nic ack chas 2 slot 1 seq 10 cookie 3
<snap> swlogd: ipmsNi msg debug1(6) cmm recv len 136
<snap> swlogd: ipmsNi msg debug1(6) cmm collect index 3 enable 1
<snap> swlogd: ipmsNi call debug1(6) collect index 3 enable 1
<snap> swlogd: ipmsNi ipms debug2(7) collect index 3 enable 1
<snap> swlogd: ipmsNi msg debug1(6) cmm rep index 3 type 1
<snap> swlogd: ipmsNi msg debug1(6) cmm set index 3 port 2/1/1
<snap> swlogd: ipmsNi call debug1(6) set index 3 port 2/1/1
<snap> swlogd: ipmsNi msg debug1(6) cmm mtu index 3 mtu 1500
<snap> swlogd: ipmsNi call debug1(6) mtu index 3 mtu 1500
<snap> swlogd: ipmsNi msg debug1(6) cmm up index 3 port 1/1/48 vlan 1 nalv 1
<snap> swlogd: ipmsNi call debug1(6) up index 3 port 1/1/48 vlan 1 nalv 1
<snap> swlogd: ipmsNi ipms debug2(7) L2: unit 0 index 3 port 48
<snap> swlogd: ipmsNi msg debug1(6) cmm trap index 3 enable 0
<snap> swlogd: ipmsNi msg debug1(6) cmm valid index 3
<snap> swlogd: ipmsNi msg debug1(6) cmm collect index 3 enable 0
<snap> swlogd: ipmsNi call debug1(6) collect index 3 enable 0
<snap> swlogd: ipmsNi ipms debug2(7) collect index 3 enable 0
<snap> swlogd: ipmsNi msg debug1(6) cmm ack seq 10
<snap> swlogd: ipmsCmm msg debug1(6) nic recv len 20
<snap> swlogd: ipmsCmm msg debug1(6) nic ack chas 1 slot 1 seq 10
<snap> swlogd: ipmsCmm msg debug1(6) nic ack chas 2 slot 1 seq 10
<snap> swlogd: ipmsCmm sub debug1(6) index 3 is now ready
<snap> swlogd: ipmsCmm sub debug1(6) flow vlan 1 dest 0.0.0.0 orig 0.0.0.0 group 239.0.0.1 host
10.0.0.1 index 0->3
<snap> swlogd: ipmsCmm msg debug1(6) nic index chas 2 slot 1 vlan 1 group 239.0.0.1 host 10.0.0.1
dest 0.0.0.0 orig 0.0.0.0 index 3
<snap> swlogd: ipmsCmm msg debug1(6) sec chas 0 flow vlan 1 group 239.0.0.1 host 10.0.0.1 dest
0.0.0.0 orig 0.0.0.0 encap 1 index 3
<snap> swlogd: ipmsCmm mip debug1(6) process
```

```
<snap> swlogd: ipmsCmm mip debug1(6) view in 0 all 1
```

An example output of a single IGMPv2 Membeship Request (IGMP receiver is located is connected to port 1/1/48 and the sender to port 2/1/1, zapping enabled)

```
-> show log swlog | grep -E "ipmsCmm|ipmsNi"
<snap> swlogd: ipmsNi cap debug1(6) pd_recv/4  src 00-00-00-00-00-01 vlan 1 stack 0 modid 0 port
48 vpn 47 vp 0 cpu 1 flood 1 pd_client 0
<snap> swlogd: ipmsNi cap debug1(6) igmp type x17 vlan 1 stack 0 port 1/1/48 group 224.0.0.2 host
192.168.1.1 sa 00-00-00-00-00-01
<snap> swlogd: ipmsNi msg debug1(6) mcm report vlan 1 stack 0 port 1/1/48 vp 0 host 192.168.1.1
sa 00-00-00-00-00-01 modid 0 devport 48
<snap> swlogd: ipmsCmm msg debug1(6) cni recv len 76
<snap> swlogd: ipmsCmm msg debug1(6) cni report vlan 1 stack 0 port 1/1/48 svp 0 host 192.168.1.1
sa 00-00-00-00-00-01 modid 0 devport 48
<snap> swlogd: ipmsCmm call debug1(6) report vlan 1 stack 0 ifindex 1/1/48 host 192.168.1.1 sa
00-00-00-00-00-01 modid 0 devport 48
<snap> swlogd: ipmsCmm sub debug1(6) translate vlan 1 stack 0 ifindex 1048(1/1/48) group
239.0.0.1
<snap> swlogd: ipmsCmm sub debug1(6) policy/4 vlan 1 ifindex 1/1/48 group 239.0.0.1 host
192.168.1.1 sa 00-00-00-00-00-01
<snap> swlogd: ipmsCmm rpt debug1(6) igmp/2 leave vlan 1 ifindex 1/1/48 host 192.168.1.1 group
239.0.0.1
<snap> swlogd: ipmsCmm rpt debug1(6) zap timer vlan 1 ifindex 1/1/48 group 239.0.0.1
<snap> swlogd: ipmsCmm call debug1(6) relay ia4_t vlan 1 len 28
<snap> swlogd: ipmsCmm msg debug1(6) sec proxy msg_report4 len 68 rem_chas 2
<snap> swlogd: ipmsCmm age debug1(6) member vlan 1 ifindex 1/1/48 group 239.0.0.1
<snap> swlogd: ipmsCmm sub debug1(6) remove vlan 1 group 239.0.0.1 host 10.0.0.1 next 1 ifindex
1/1/48
<snap> swlogd: ipmsCmm obj debug1(6) fabric del vlan 1 group 239.0.0.1 host 10.0.0.1 next 1
ifindex 1/1/48
<snap> swlogd: ipmsCmm obj debug1(6) thread del vlan 1 group 239.0.0.1 host 10.0.0.1 next 1
<snap> swlogd: ipmsCmm obj debug1(6) channel del vlan 1 group 239.0.0.1
<snap> swlogd: ipmsCmm call debug1(6) havlan check vlan 1 group 239.0.0.1 ifindex 1/1/48 allow 0
<snap> swlogd: ipmsCmm obj debug1(6) member del vlan 1 ifindex 1/1/48 group 239.0.0.1
<snap> swlogd: ipmsCmm sub debug1(6) settle fwdvecs 0 v4flows 1 v6flows 0
<snap> swlogd: ipmsCmm sub debug1(6) flow vlan 1 dest 0.0.0.0 orig 0.0.0.0 group 239.0.0.1 host
10.0.0.1 index 3->0
<snap> swlogd: ipmsCmm msg debug1(6) nic undex chas 2 slot 1 vlan 1 group 239.0.0.1 host 10.0.0.1
dest 0.0.0.0 orig 0.0.0.0
<snap> swlogd: ipmsCmm msg debug1(6) sec chas 0 flow vlan 1 group 239.0.0.1 host 10.0.0.1 dest
0.0.0.0 orig 0.0.0.0 encap 1 index 0
<snap> swlogd: ipmsCmm mip debug1(6) process
<snap> swlogd: ipmsCmm mip debug1(6) view in 0 all 1
<snap> swlogd: ipmsCmm age debug1(6) fwdvec mcindex 3 inuse 0
<snap> swlogd: ipmsCmm obj debug1(6) fwdvec del mcindex 3 vlan 1 ifindex 2/1/1 fwds 1 trap 0
<snap> swlogd: ipmsCmm obj debug1(6) mcindex del index 3
<snap> swlogd: ipmsCmm msg debug1(6) nic clear chas 1 slot 1 index 3
<snap> swlogd: ipmsCmm msg debug1(6) nic clear chas 2 slot 1 index 3
<snap> swlogd: ipmsCmm obj debug1(6) seq del id 12 cookie 3 refcnt 0
<snap> swlogd: ipmsCmm sub debug1(6) settle fwdvecs 0 v4flows 0 v6flows 0
<snap> swlogd: ipmsNi msg debug1(6) cmm recv len 20
<snap> swlogd: ipmsNi msg debug1(6) cmm clear index 3
<snap> swlogd: ipmsNi call debug1(6) clear index 3
```

14.3. **Advanced troubleshooting**

A more detailed version of "show ip multicast group"

```
-> debug ip multicast member
Total 1 Members
Group Address/  VLAN  Port      Mode     Count  Life  Query  Count  V   V1    V2
Source Address
---------------+-----+---------+--------+------+-----+------+------+---+-----+-----
239.0.0.1       1     1/1/48    exclude  3      259   0      0      2   0     259
```

A more detailed version of "show ip multicast flow"

```
-> debug ip multicast flow

Total 1 Flows
indexes inuse 1  max 8189
Group Address/  Host Address/   Next Address    VLAN/ Port      Index  Chas_ID
Dest Address    Orig Address
                                               Next
---------------+---------------+---------------+-----+---------+------+-------
239.0.0.1       10.0.0.1        0.0.0.0         1     2/1/1     3      2
0.0.0.0         0.0.0.0
                                                1
```

An example output of "debug ip multicat channel":

```
-> debug ip multicast channel
Total 1 Channels
Group Address/  VLAN  Mode     Count  Life  V   V1    V2
Source Address
---------------+-----+--------+------+-----+---+-----+-----
239.0.0.1       1     exclude  6      194   2   0     194
```

Display IP interface configuration related to multicast:

```
-> debug ip multicast interface
Total 3 Interfaces
IfIndex   Host Address    Mac Address         VLAN  VRF  Other  Query  Count
---------+---------------+-------------------+-----+----+------+------+------
13600001  192.168.10.253  00-00-00-00-00-00   10    0    0      0      0
13600002  192.168.100.254 e8-e7-32-ab-17-bd   100   0    0      0      0
13604125  127.0.0.1       00-00-00-00-00-00   0     0    0      0      0
```

A more detailed version of "show ip muticast vlan":

```
-> debug ip multicast vlan 1
Routing                                         = disabled,
Turning                                         = disabled,
Elected                                         = false,
Group Membership Interval (seconds)             = 260,
Querier Interval (seconds)                      = 255,
Startup Query Interval (seconds)                = 31,
Last Member Query Time (seconds)                = 2,
Unsolicited Report Time (seconds)               = 1,
Cvg Query Response Interval (tenths of seconds) = 25,
Cvg Startup Query Interval (seconds)            = 7,
IGMPv1 Querier Present (seconds)                = 0,
IGMPv2 Querier Present (seconds)                = 241
```

Display IP multicast statistics:

```
-> debug ip multicast stats
Global RX Statistics
V1 Reports              =          0 | V1 Queries              =          0
V2 Reports              =         19 | V2 Queries              =          9
V2 Leaves               =          3 |
V3 Reports              =          2 | V3 Queries              =          0
PIM Hellos              =          0 | DVMRP Probes            =          0
Global TX Statistics
V1 Reports              =          0 | V1 Queries              =          0
V2 Reports              =          6 | V2 Queries              =         27
V2 Leaves               =          1 |
V3 Reports              =          0 | V3 Queries              =          0
```

14.4. **Troubleshooting in the Maintenance Shell**

Warning: Maintenance Shell commands should only be used by Alcatel-Lucent personnel or under the direction of Alcatel-Lucent. Misuse or failure to follow procedures that use Maintenance Shell commands in this guide correctly can cause lengthy network down time and/or permanent damage to hardware.

Verify the software forwarding limit:

```
TOR #-> debug $(pidof ipmscmm) 'p ipms::bcm_max'
```

```
[Thread debugging using libthread_db enabled]
0x0f990c2c in ___newselect_nocancel () from /lib/tls/libc.so.6
$1 = 2048
```

Change the software forwarding limit:

```
TOR #-> debug $(pidof ipmscmm) 'set ipms::bcm_max=4096'
[Thread debugging using libthread_db enabled]
0x0f990c2c in ___newselect_nocancel () from /lib/tls/libc.so.6
```

14.5. **Troubleshooting in bShell**

OS6900 and OS10K tables

MC_CONTROL_5

Display the MC_CONTROL_5 register:

Note: Only applicable to AOS7 hardware platform

```
BCM.0> getreg MC_CONTROL_5
MC_CONTROL_5.ipipe0[0xc180609]=0x2001000:
<SHARED_TABLE_L2MC_SIZE=0x1000,SHARED_TABLE_IPMC_SIZE=0x1000>
```

L2MC table

Note: Only applicable to AOS7 hardware platform

If the destination MAC address is a multicast address, then the result of the destination lookup is a 10-bit index (L2MC_PTR) into this table. The result of the direct index into the L2 multicast table is a bitmap that indicates which ports on the local switch should receive the packet. The MC Port Bitmap is qualified with the VLAN bitmap. The MC Port bitmap is picked up from the L2MC table.
Default "L2MC Table":

```
BCM.0> d chg l2mc
L2MC.ipipe0[0]: <VALID=1,PORT_BITMAP_W1=0x400000,PORT_BITMAP=0x0040000000000000,>
L2MC.ipipe0[1]:
<VALID=1,PORT_BITMAP_W1=0x400000,PORT_BITMAP_W0=1,PORT_BITMAP=0x0040000000000001,EVEN_PARITY=1>
L2MC.ipipe0[3]:
<VALID=1,PORT_BITMAP_W1=0xffffff,PORT_BITMAP_W0=0xfffffffe,PORT_BITMAP=0x00fffffffffffffe,>
```

L3_IPMC table

Note: Only applicable to AOS7 hardware platforms

The index into this table is specified by L3MC_INDEX bits in the L3_ENTRY tables.

Default "L3_IPMC Table":

```
BCM.0> d l3_ipmc
L3_IPMC.ipipe0[0]:
<VALID=1,REMOVE_SGLP_FROM_L3_BITMAP=0,L3_BITMAP_W1=0,L3_BITMAP_W0=0,L3_BITMAP=0x0000000000000000,
L2_BITMAP_W1=0,L2_BITMAP_W0=0,L2_BITMAP=0x0000000000000000,HIGIG_TRUNK_OVERRIDE_PROFILE_PTR=0,EVE
N_PARITY=1>
L3_IPMC.ipipe0[1]:
<VALID=1,REMOVE_SGLP_FROM_L3_BITMAP=0,L3_BITMAP_W1=0,L3_BITMAP_W0=0,L3_BITMAP=0x0000000000000000,
L2_BITMAP_W1=0xffffff,L2_BITMAP_W0=0xfffffffe,L2_BITMAP=0x00fffffffffffffe,HIGIG_TRUNK_OVERRIDE_P
ROFILE_PTR=0,EVEN_PARITY=0>
L3_IPMC.ipipe0[2]:
<VALID=1,REMOVE_SGLP_FROM_L3_BITMAP=0,L3_BITMAP_W1=0,L3_BITMAP_W0=0,L3_BITMAP=0x0000000000000000,
L2_BITMAP_W1=0,L2_BITMAP_W0=1,L2_BITMAP=0x0000000000000001,HIGIG_TRUNK_OVERRIDE_PROFILE_PTR=0,EVE
N_PARITY=0>
L3_IPMC.ipipe0[3]:
<VALID=1,REMOVE_SGLP_FROM_L3_BITMAP=0,L3_BITMAP_W1=0,L3_BITMAP_W0=0,L3_BITMAP=0x0000000000000000,
L2_BITMAP_W1=0xc00000,L2_BITMAP_W0=4,L2_BITMAP=0x00c0000000000004,HIGIG_TRUNK_OVERRIDE_PROFILE_PT
R=0,EVEN_PARITY=0>
```

A new entry is added to this table when a new IPMS forwarding entry is created (an example):

```
BCM.0> d l3_ipmc
...
L3_IPMC.ipipe0[3]:
<VALID=1,REMOVE_SGLP_FROM_L3_BITMAP=0,L3_BITMAP_W1=0,L3_BITMAP_W0=0,L3_BITMAP=0x0000000000000000,
L2_BITMAP_W1=0xc00000,L2_BITMAP_W0=4,L2_BITMAP=0x00c0000000000004,HIGIG_TRUNK_OVERRIDE_PROFILE_PT
R=0,EVEN_PARITY=0>
```

OS6860 tables

L3_ENTRY table

Note: Only applicable to AOS8 hardware platform

L3_ENTRY_2 table is empty by default. A new entry is created while adding a new entry in the multicast source table (a forwarding entry doesn't have to exist, an example 239.0.0.1 stream):

```
BCM.0> d l3_entry_2
L3_ENTRY_2.ism0[1822]: <VALID_1=1,VALID_0=1,KEY
_TYPE_1=6,KEY_TYPE_0=6,IPV4MC:VRF_ID=0,IPV4MC:VLAN_ID=0x64,
 IPV4MC:TRILL_NETWORK_RECEIVERS_PRESENT=0,IPV4MC:SOURCE_IP_ADDR=0xa1dc915,IPV4MC:RPE=0,
IPV4MC:RESERVED_209_144=0x000000000000000000,IPV4MC:RESERVED_104_97=0,IPV4MC:PRI=0,IPV4MC:L3_IIF=
0x64,
 IPV4MC:L3MC_INDEX_RESERVED=0,IPV4MC:L3MC_INDEX=0,IPV4MC:KEY_0=0x00c9de000002143b922a0006,
IPV4MC:HASH_LSB=1,IPV4MC:GROUP_IP_ADDR=0xef000001,IPV4MC:DUMMY_0=0,IPV4MC:DST_DISCARD=0,IPV4MC:DA
TA_1=0x0000000000,
 IPV4MC:CLASS_ID=0,HIT_BITS=1,HIT_2=0,HIT_1=0,HIT_0=1,EVEN_PARITY_1=1,EVEN_PARITY_0=1,
ENTRY_2_FROM_ENTRY_1_PART1=0x000000000000000000000000001b,ENTRY_2_FROM_ENTRY_1_PART0=0x0000032778
00000850ee48a8001b,>
```

L3_IPMC table

Note: Only applicable to AOS8 hardware platforms

Default "L3_IPMC Table":

```
BCM.0> d l3_ipmc
L3_IPMC.ipipe0[0]:
<VALID=1,REMOVE_SGLP_FROM_L3_BITMAP=0,L3_BITMAP_W1=0,L3_BITMAP_W0=0,L3_BITMAP=0x0000000000000000,
L2_BITMAP_W1=0,L2_BITMAP_W0=0,L2_BITMAP=0x0000000000000000,HIGIG_TRUNK_OVERRIDE_PROFILE_PTR=0,EVE
N_PARITY=1>
L3_IPMC.ipipe0[1]:
<VALID=1,REMOVE_SGLP_FROM_L3_BITMAP=0,L3_BITMAP_W1=0,L3_BITMAP_W0=0,L3_BITMAP=0x0000000000000000,
L2_BITMAP_W1=0xffffff,L2_BITMAP_W0=0xfffffffe,L2_BITMAP=0x00fffffffffffffe,HIGIG_TRUNK_OVERRIDE_P
ROFILE_PTR=0,EVEN_PARITY=0>
L3_IPMC.ipipe0[2]:
<VALID=1,REMOVE_SGLP_FROM_L3_BITMAP=0,L3_BITMAP_W1=0,L3_BITMAP_W0=0,L3_BITMAP=0x0000000000000000,
L2_BITMAP_W1=0,L2_BITMAP_W0=1,L2_BITMAP=0x0000000000000001,HIGIG_TRUNK_OVERRIDE_PROFILE_PTR=0,EVE
N_PARITY=0>
A new entry is added to this table when a new IPMS forwarding entry is creatred (an example):
BCM.0> d l3_ipmc
...
L3_IPMC.ipipe0[3]:
<VALID=1,REMOVE_SGLP_FROM_L3_BITMAP=0,L3_BITMAP_W1=0,L3_BITMAP_W0=0,L3_BITMAP=0x0000000000000000,
L2_BITMAP_W1=0xc00000,L2_BITMAP_W0=0x1000,L2_BITMAP=0x00c0000000001000,HIGIG_TRUNK_OVERRIDE_PROFI
LE_PTR=0,EVEN_PARITY=0>
```

15. Troubleshooting IP Multicast Routing (IPMR)

Checklist

- IP interfaces for all concerned VLANs must be added into multicast routing context
- The IP interface used as Rendezvous Point must be added as PIM interface
- In case of PIM Sparse Mode a static or candidate Rendezvous Point must be configured
- Multicast sources must use IP addresses matching the subnet of the VLAN
- TTL in packets generated by multicast sources must be sufficient to reach all destinations

Summary of the commands in this chapter is listed here:

```
show configuration snapshot ip ipms ipmr
show ip multicast source | head
show ip multicast group | head
show ip multicast forward | head
show ip pim interface
show ip pim sgroute | head -n 14
show ip pim groute | head
debug ip multicast flow
show c xe11
```

15.1. Introduction

IPMS and Multicast Routing Limitations

IP Multicast routing and switch control plane traffic are done in the software path, such as IGMP, MLD, PIM Join/Prune, Hello – DVMRP, PIM-DM, PIM-SSM, etc. IP multicast will be forwarded in hardware mode in switched or routed networks, except register packets and DVMRP tunneling packets.

15.2. Minimum working configuration

PIM Sparse Mode

```
-> show configuration snapshot ip ipms ipmr
! IP:
ip interface "Loopback0" address 192.168.254.1
ip interface "vlan10" address 192.168.10.1 mask 255.255.255.0 vlan 10 ifindex 1
ip interface "vlan20" address 192.168.20.1 mask 255.255.255.0 vlan 20 ifindex 2
! IPMS:
ip multicast admin-state enable
! IP Multicast:
ip load pim
ip pim interface "vlan10"
ip pim interface "vlan20"
ip pim interface "Loopback0"
ip pim static-rp 224.0.0.0/4 192.168.254.1
ip pim sparse admin-state enable
ip pim dense admin-state disable
ipv6 pim sparse admin-state disable
ipv6 pim dense admin-state disable
! DVMRP:
! IPMR:
```

15.3. Basic Troubleshooting

```
-> show ip multicast source | head
Total 100 Sources
Group Address   Host Address    Tunnel Address  VLAN  Port
---------------+---------------+---------------+-----+---------
239.0.0.1       192.168.1.19    0.0.0.0          1    1/11
239.0.0.2       192.168.1.19    0.0.0.0          1    1/11
239.0.0.3       192.168.1.19    0.0.0.0          1    1/11
239.0.0.4       192.168.1.19    0.0.0.0          1    1/11
239.0.0.5       192.168.1.19    0.0.0.0          1    1/11
-> show ip multicast group | head
Total 100 Groups
Group Address   Source Address  VLAN  Port      Mode     Static  Count  Life
---------------+---------------+-----+---------+--------+-------+------+-----
239.0.0.1       0.0.0.0          2    1/12      exclude  no      16115  259
239.0.0.2       0.0.0.0          2    1/12      exclude  no      9203   259
239.0.0.3       0.0.0.0          2    1/12      exclude  no      9237   259
239.0.0.4       0.0.0.0          2    1/12      exclude  no      9097   259
239.0.0.5       0.0.0.0          2    1/12      exclude  no      9192   259
-> show ip multicast forward | head
Total 100 Forwards
                                                  Ingress         Egress
Group Address   Host Address    Tunnel Address  VLAN  Port      VLAN  Port
---------------+---------------+---------------+-----+---------+-----+---------
239.0.0.1       192.168.1.19    0.0.0.0          1    1/11       2    1/12
239.0.0.2       192.168.1.19    0.0.0.0          1    1/11       2    1/12
239.0.0.3       192.168.1.19    0.0.0.0          1    1/11       2    1/12
239.0.0.4       192.168.1.19    0.0.0.0          1    1/11       2    1/12
-> show ip pim interface
Total 3 Interfaces
Interface Name                   IP Address      Designated      Hello    J/P      Oper     BFD
                                                 Router          Interval Interval Status
Status
--------------------------------+---------------+---------------+--------+--------+--------+-----
---
vlan10                           192.168.10.1    192.168.10.1    30       60       enabled
disabled
vlan20                           192.168.20.1    192.168.20.1    30       60       enabled
disabled
Loopback0                        192.168.254.1   192.168.254.1   30       60       enabled
disabled
-> show ip pim sgroute | head -n 14

Legend: Flags: D = Dense, S = Sparse, s = SSM Group,
               L = Local, R = RPT, T = SPT, F = Register,
               P = Pruned, O = Originator
Total 100 (S,G)
Source Address  Group Address   RPF Interface                    Upstream Neighbor UpTime
Flags
---------------+---------------+--------------------------------+-----------------+-----------+--
------
192.168.1.19    239.0.0.1       vlan10                                             00h:20m:56s
STL
192.168.1.19    239.0.0.2       vlan10                                             00h:15m:19s
STL
192.168.1.19    239.0.0.3       vlan10                                             00h:15m:19s
STL
192.168.1.19    239.0.0.4       vlan10                                             00h:15m:19s
STL
192.168.1.19    239.0.0.5       vlan10                                             00h:15m:19s
STL
-> show ip pim groute | head
Total 100 (*,G)
Group Address   RP Address      RPF Interface                    Upstream Neighbor UpTime
Mode
---------------+---------------+--------------------------------+-----------------+-----------+--
----
239.0.0.1       192.168.254.1                                                      00h:28m:33s
ASM
```

```
239.0.0.2       192.168.254.1                                                      00h:15m:40s
ASM
239.0.0.3       192.168.254.1                                                      00h:15m:40s
ASM
239.0.0.4       192.168.254.1                                                      00h:15m:40s
ASM
239.0.0.5       192.168.254.1                                                      00h:15m:40s
ASM
```

15.4. **Advanced Troubleshooting**

```
-> debug ip multicast flow
Total 10 Flows
MCIDX inuse 10  max 2045
Group Address/  Host Address/   Next Address    VLAN/ Port      Index  Chas_ID
Dest Address    Orig Address
                                                Next
---------------+---------------+---------------+-----+---------+------+-------
239.0.0.1       192.168.1.19    0.0.0.0         1     1/11      4      0
0.0.0.0         0.0.0.0
                                                1
                                                2
239.0.0.2       192.168.1.19    0.0.0.0         1     1/11      3      0
0.0.0.0         0.0.0.0
                                                1
                                                2
239.0.0.3       192.168.1.19    0.0.0.0         1     1/11      5      0
0.0.0.0         0.0.0.0
...
```

15.5. **Troubleshooting in bShell**

Verifying if multicast multicast traffic is forwarded (PERQ_PKT(0) increments on egress port, this counter doesn't increment if TTL on ingress is equal 0 or 1):

```
BCM.0> show c xe11
RDBGC0.xe11             :                  1,155                  +1               1/s
RDBGC1.xe11             :                  1,155                  +1               1/s
R64.xe11                :                  1,159                  +1               1/s
RPKT.xe11               :                  1,161                  +1               1/s
RMCA.xe11               :                  1,155                  +1               1/s
RPOK.xe11               :                  1,161                  +1               1/s
RBYT.xe11               :                 74,380                 +64              54/s
T64.xe11                :                223,568              +1,183           1,001/s
T127.xe11               :                    941                  +1               1/s
TPOK.xe11               :                224,509              +1,184           1,002/s
TPKT.xe11               :                224,509              +1,184           1,002/s
TMCA.xe11               :                224,497              +1,184           1,002/s
TBYT.xe11               :             14,375,022             +75,780          64,106/s
BCM.0> show c xe11
PERQ_PKT(0).xe11        :                223,494              +1,182           1,000/s
PERQ_BYTE(0).xe11       :             14,303,616             +75,648          63,994/s
UC_PERQ_PKT(9).xe11     :                  1,003                  +1               1/s
UC_PERQ_BYTE(9).xe11    :                 69,738                 +68              58/s
```

16. Troubleshooting 802.1X

Summary of the commands in this chapter is listed here:

show unp user
show unp edge-user details

This section concerns the OmniSwitch 6860 running AOS 8

1) Verify the configuration as there are multiple profiles and associations to create:

RADIUS server to aaa profile:

```
aaa radius-server "clearpass" host 172.26.61.6
aaa profile "clearpass-aaa-profile"
aaa profile "clearpass-aaa-profile" device-authentication 802.1x "clearpass"
aaa profile "clearpass-aaa-profile" accounting 802.1x "clearpass"
```

Edge profile and aaa profile to edge template,

```
unp edge-profile clearpass-ep
unp vlan-mapping edge-profile clearpass-ep vlan 21
unp edge-template clearpass-et
unp edge-template clearpass-et 802.1x-authentication enable
unp edge-template clearpass-et 802.1x-authentication pass-alternate edge-profile clearpass-ep
unp edge-template clearpass-et aaa-profile clearpass-aaa-profile
```

Edge template to the port

```
unp port 1/1/45 port-type edge
unp port 1/1/45 edge-template clearpass-et
```

2) Test the RADIUS server:

RADIUS test tool allows the user to test the RADIUS server reachability from the OmniSwitch. Use this command to start the authentication or accounting test for the specified user name and password.

```
aaa test-radius-server clearpass type authentication user alcatel password alcatel123 method pap
Testing Radius Server <172.26.61.6/clearpass>
Access-Accept from 172.26.61.6 Port 1812 Time: 212 ms
Returned Attributes
    Filter-ID = employee
```

Be aware that the authentication method can only be MD5 or PAP, the server may not be configured for those methods so additional RADIUS server configuration might be required..
3) Check the authentication status

```
-> show unp  user
Port   Username Mac address        IP             Vlan  Profile       Type  Status  Source
1/1/47 julien    00:15:17:51:d3:8f  192.168.21.13  21    clearpass-ep  Edge  Active  Local
Total users : 1

-> show unp edge-user details
Port: 1/1/47
    MAC-Address: 00:15:17:51:d3:8f
      Access Timestamp                = 02/21/2014 00:46:10,
      User Name                       = julien,
      IP-Address                      = 192.168.21.13,
      Vlan                            = 21,
      Authentication Type             = 802.1x,
      Authentication Status           = Authenticated,
      Authentication Failure Reason   = -,
      Authentication Retry Count      = 0,
      Authentication Server IP Used   = 172.26.61.6,
      Authentication Server Used      = clearpass,
      Server Reply-Message            = -,
      Profile                         = clearpass-ep,
      Profile Source                  = Auth - Pass Alternate UNP,
      Profile From Auth Server        = -,
      Classification Profile Rule     = -,
      Role                            = -,
      Role Source                     = -,
      User Role Rule                  = -,
      Restricted Access               = No,
      Location Policy Status          = -,
      Time Policy Status              = -,
      Captive-Portal Status           = -,
      QMR Status                      = Passed,
      Redirect Url                    = -,
      SIP Call Type                   = Not in a call,
      SIP Media Type                  = None,
      Applications                    = None
Total users : 1 Port: 1/1/47
4) Always check also on the server side, you will find most of the log for the issues there :

Example with Clearpass with a test with "test-radius-server" without pap configured :

Error Code:    216
Error Category:       Authentication failure
Error Message:        User authentication failed
Alerts for this Request
RADIUS  Cannot select appropriate authentication method
2015-01-14 11:22:39,160       [Th 37 Req 4 SessId R00000004-01-54b6517e] ERROR
RadiusServer.Radius - rlm_auth_check: Auth-Type not set or authentication methods have not been
configured. Rejecting it.
```

5) Packet captures can always be useful:

If the logs of the RADIUS server are not helpful, a packet capture between the client and/or the uplink to the RADIUS server may be useful.
On the client side, check that the client or the switch initiates the EAP session:

- If the client initiates the EAP session but doesn't get a reply, the switch may not have a complete setup to manage 802.1x.
- If the switch initiates the EAP session, but the client does not reply, the client likely does not have a completely configured 802.1x service (check "Wired AutoConfig" service for Windows).

On the uplink to the server check the exchange between the switch and the RADIUS server. If you can ping the server but the switch get not reply, you may have a firewall issue, RADIUS uses ports 1812 and 1813 by default.

17. Troubleshooting Universal Network Profiles (UNP)

Summary of the commands in this chapter is listed here:

```
show unp user
d port 1 1
mod port 1 1 PORT_VID=1
```

17.1. Troubleshooting in bShell

Example of one trouble shooting scenario:
Issue: UNP classification on SAP access port does not happen randomly on OS6900-X40
In this case, PORT_VID on port 1 value was set as "0xfff" at BCM. This causes the packet not going to CPU for UNP classification. PORT_VID on port 19 was fine and had no issue for the user classification.

Go to maintenance shell / bShell

```
BCM.0> d port 1 1
PORT.ipipe0[1]:
<VT_MISS_DROP=0,VT_KEY_TYPE_USE_GLP=1,VT_KEY_TYPE_2_USE_GLP=1,VT_KEY_TYPE_2=5,VT_KEY_TYPE=4,VT_EN
ABLE=0,
VNTAG_ACTIONS_IF_PRESENT=0,VNTAG_ACTIONS_IF_NOT_PRESENT=0,VLAN_PROTOCOL_DATA_INDEX=1,VLAN_PRECEDE
NCE=0,VINTF_CTR_IDX=0,
VFP_PORT_GROUP_ID=1,VFP_ENABLE=1,V6L3_ENABLE=1,V6IPMC_L2_ENABLE=0,V6IPMC_ENABLE=0,V4L3_ENABLE=1,V
4IPMC_L2_ENABLE=0,V4IPMC_ENABLE=0,
USE_PORT_TABLE_GROUP_ID=0,USE_IVID_AS_OVID=0,USE_INNER_PRI=0,URPF_MODE=0,URPF_DEFAULTROUTECHECK=0
,TX_DEST_PORT_ENABLE=0,TX_DEST_PORT=0,
TRUST_INCOMING_VID=1,TRUST_DSCP_V6=0,TRUST_DSCP_V4=0,TRUST_DOT1P_PTR=0,TRILL_ENABLE=0,TAG_ACTION_
PROFILE_PTR=0,SUBNET_BASED_VID_ENABLE=1,
RTAG7_PORT_LBN=0,RTAG7_HASH_CFG_SEL_TRUNK=0,RTAG7_HASH_CFG_SEL_TRILL_ECMP=0,RTAG7_HASH_CFG_SEL_LB
ID=0,RTAG7_HASH_CFG_SEL_HIGIG_TRUNK=0,
RTAG7_HASH_CFG_SEL_ECMP=0,RESERVED_1=0,RESERVED_0=0,REMOVE_HG_HDR_SRC_PORT=0,REMOTE_CPU_EN=0,PVLA
N_ENABLE=0,PROTOCOL_PKT_INDEX=2,
PROHIBITED_DOT1P=0,PRI_MAPPING=0xfac688,PORT_VID=0xfff,PORT_TYPE=0,PORT_PRI=0,PORT_OPERATION=0,PO
RT_DIS_UNTAG=0,PORT_ DIS_TAG=0,
//snip//

BCM.0> d port 19 1
PORT.ipipe0[19]:
<VT_MISS_DROP=0,VT_KEY_TYPE_USE_GLP=1,VT_KEY_TYPE_2_USE_GLP=1,VT_KEY_TYPE_2=5,VT_KEY_TYPE=4,VT_EN
ABLE=0,
VNTAG_ACTIONS_IF_PRESENT=0,VNTAG_ACTIONS_IF_NOT_PRESENT=0,VLAN_PROTOCOL_DATA_INDEX=0x13,VLAN_PREC
EDENCE=0,VINTF_CTR_IDX=0,
VFP_PORT_GROUP_ID=1,VFP_ENABLE=1,V6L3_ENABLE=1,V6IPMC_L2_ENABLE=0,V6IPMC_ENABLE=0,V4L3_ENABLE=1,V
4IPMC_L2_ENABLE=0,V4IPMC_ENABLE=0,
USE_PORT_TABLE_GROUP_ID=0,USE_IVID_AS_OVID=0,USE_INNER_PRI=0,URPF_MODE=0,URPF_DEFAULTROUTECHECK=0
,TX_DEST_PORT_ENABLE=0,TX_DEST_PORT=0,
TRUST_INCOMING_VID=1,TRUST_DSCP_V6=0,TRUST_DSCP_V4=0,TRUST_DOT1P_PTR=0,TRILL_ENABLE=0,TAG_ACTION_
PROFILE_PTR=3,SUBNET_BASED_VID_ENABLE=0,
RTAG7_PORT_LBN=0,RTAG7_HASH_CFG_SEL_TRUNK=0,RTAG7_HASH_CFG_SEL_TRILL_ECMP=0,RTAG7_HASH_CFG_SEL_LB
ID=0,RTAG7_HASH_CFG_SEL_HIGIG_TRUNK=0,
RTAG7_HASH_CFG_SEL_ECMP=0,RESERVED_1=0,RESERVED_0=0,REMOVE_HG_HDR_SRC_PORT=0,REMOTE_CPU_EN=0,PVLA
N_ENABLE=0,PROTOCOL_PKT_INDEX=2,
PROHIBITED_DOT1P=0,PRI_MAPPING=0xfac688,PORT_VID=1,PORT_TYPE=0,PORT_PRI=0,PORT_OPERATION=0,PORT_D
IS_UNTAG=0,PORT_DIS_TAG=0,
 //snip//
```

Modified the parameter of PORT_VID in BCM. After this changes, user entry on port 1 is updated correctly (as same as port 19)

```
BCM.0> mod port 1 1 PORT_VID=1
OS6900-3> show unp user
                                            User
Learning
Port   Username           Mac address         IP              Vlan  UNP
Status   Source
------+-----------------+-----------------+---------------+----+--------------------------------
+------+----------
1/1/1  00:13:72:7a:f3:73 00:13:72:7a:f3:73 192.168.1.253   4095 spb1
Active    Local
1/1/19 00:13:72:7a:11:11 00:13:72:7a:11:11 12.13.14.15     4095 spb2
Active    Local
```

18. Troubleshooting SNMP

The Simple Network Management Protocol (SNMP) is an application-layer protocol that allows communication between SNMP managers and SNMP agents on an IPv4 as well as on an IPv6 network.

Network administrators use SNMP to monitor network performance and to manage network resources.
In This Chapter

- “Troubleshooting SNMP on OmniSwitch OS6900/OS10K/OS6860”
- “SNMP Security”
- “SNMP Statistics”
- “Debug Troubleshooting”

Summary of the commands in this chapter is listed here:

show configuration snapshot snmp
show snmp station
show user public
show snmp-trap filter-ip
show snmp-trap config
show snmp community-map
show user snmptest
show snmp security
show snmp statistics
debug snmp data community
debug snmp data user
debug trap counts

18.1. Troubleshooting SNMP on OmniSwitch OS6900/OS10K/OS6860 series

The User ID and password community string must be configured; make sure that these variables are correct.

To view the SNMP configuration, use the **show configuration snapshot snmp** command:

```
-> show configuration snapshot snmp

! Trap Manager:
snmp station 10.100.10.21 1162 "public" v2 enable

! SNMP:
snmp security no-security
snmp authentication-trap enable
snmp community-map mode enable
snmp community-map "public" user "public" enable
```

SNMP Network Management Station (NMS) is a workstation configured to receive SNMP traps from the switch. The OmniSwitch supports SNMP v1, v2, and v3. The most often mistake is when the wrong workstation IP address is configured. The workstation can ping the switch, but no traps are being received.

To verify the SNMP Management Station, use the **show snmp station** command:

```
-> show snmp station
ipAddress/udpPort                                    status    protocol user
---------------------------------------------------+---------+--------+-------
10.100.10.21/1162                                     enable    v2       public
```

Verify the user account name and the authentication type for that user by using the **show user** command from the CLI:

```
-> show user public
User name = public,
  Password expiration     = None,
  Password allow to be modified date     = None,
  Account lockout     = None,
  Password bad attempts     = 0,
  Read Only for domains   = None,
  Read/Write for domains  = All ,
  Snmp allowed     = YES,
  Snmp authentication     = NONE,
  Snmp encryption     = NONE
  Console-Only    = Disabled
```

Verify whether or not trap filters are configured on the switch. If the switch is configured with SNMP trap filters, the switch will not pass the specified traps through to the SNMP management station. All other SNMP traps will be passed through.

To verify the SNMP trap filter configuration, use the **show snmp-trap filter-ip** command:

```
-> show snmp-trap filter-ip
ipAddress                                 trapId list
-----------------------------------------+------------------------------------
10.100.10.21                              no filter
```

To display SNMP trap information, including trap ID numbers, trap names, command families, and absorption rate, use the **show snmp-trap config** command. This command also displays the enabled/disabled status of SNMP absorption.

For example:

```
-> show snmp-trap config

Absorption service : enabled
Traps to WebView : enabled

id trap name                              family          absorption
--+------------------------------------+----------------+------------
 0 coldStart                               chassis         15 seconds
 1 warmStart                            chassis         15 seconds
 2 linkDown                             interface       15 seconds
 3 linkUp                                   interface       15 seconds
 4 authenticationFailure                 snmp            15 seconds
 5 entConfigChange                      module          15 seconds
 6 policyEventNotification              qos             15 seconds
 7 chassisTrapsStr                      chassis         15 seconds
 8 chassisTrapsAlert                    chassis         15 seconds
 9 chassisTrapsStateChange              chassis         15 seconds
10 chassisTrapsMacOverlap               module          15 seconds
11 vrrpTrapNewMaster                    vrrp            15 seconds
12 vrrpTrapAuthFailure                  vrrp            15 seconds
13 healthMonModuleTrap                  health          15 seconds
14 healthMonPortTrap                    health          15 seconds
15 healthMonCmmTrap                     health          15 seconds
16 bgpEstablished                       bgp             15 seconds
17 bgpBackwardTransition                bgp             15 seconds
18 esmDrvTrapDropsLink                  interface       15 seconds
19 portViolationTrap                    interface       15 seconds
20 dvmrpNeighborLoss                    ipmr            15 seconds
21 dvmrpNeighborNotPruning              ipmr            15 seconds
22 risingAlarm                          rmon            15 seconds
23 fallingAlarm                         rmon            15 seconds
24 stpNewRoot                           stp             15 seconds
```

```
25 stpRootPortChange                    stp             15 seconds
26 mirrorConfigError                    pmm             15 seconds
27 mirrorUnlikeNi                       pmm             15 seconds
28 slbTrapOperStatus                    loadbalancing   15 seconds
29 sessionAuthenticationTrap            session         15 seconds
30 trapAbsorptionTrap                   none            no
31 alaDoSTrap                           ip              15 seconds
32 ospfNbrStateChange                   ospf            15 seconds
33 ospfVirtNbrStateChange               ospf            15 seconds
34 lnkaggAggUp                          linkaggregation 15 seconds
35 lnkaggAggDown                        linkaggregation 15 seconds
36 lnkaggPortJoin                       linkaggregation 15 seconds
37 lnkaggPortLeave                      linkaggregation 15 seconds
38 lnkaggPortRemove                     linkaggregation 15 seconds
39 monitorFileWritten                   pmm             15 seconds
40 alaVrrp3TrapProtoError               vrrp            15 seconds
41 alaVrrp3TrapNewMaster                vrrp            15 seconds
42 chassisTrapsPossibleDuplicateMac     chassis         15 seconds
43 lldpRemTablesChange                  aip             15 seconds
44 pimNeighborLoss                      ipmr            15 seconds
45 pimInvalidRegister                   ipmr            15 seconds
46 pimInvalidJoinPrune                  ipmr            15 seconds
47 pimRPMappingChange                   ipmr            15 seconds
48 pimInterfaceElection                 ipmr            15 seconds
49 pimBsrElectedBSRLostElection         ipmr            15 seconds
50 pimBsrCandidateBSRWinElection        ipmr            15 seconds
51 lpsViolationTrap                     bridge          15 seconds
52 lpsPortUpAfterLearningWindowExpiredT bridge          15 seconds
53 lpsLearnTrap                         bridge          15 seconds
54 gvrpVlanLimitReachedEvent            bridge          15 seconds
55 alaNetSecPortTrapAnomaly             netsec          15 seconds
56 alaNetSecPortTrapQuarantine          netsec          15 seconds
57 ifMauJabberTrap                      interface       15 seconds
58 udldStateChange                      interface       15 seconds
59 ndpMaxLimitReached                   ip              15 seconds
60 ripRouteMaxLimitReached              rip             15 seconds
61 ripngRouteMaxLimitReached            ripng           15 seconds
62 alaErpRingStateChanged               bridge          15 seconds
63 alaErpRingMultipleRpl                bridge          15 seconds
64 alaErpRingRemoved                    bridge          15 seconds
65 ntpMaxAssociation                    ntp             15 seconds
66 ddmTemperatureThresholdViolated      interface       15 seconds
67 ddmVoltageThresholdViolated          interface       15 seconds
68 ddmCurrentThresholdViolated          interface       15 seconds
69 ddmTxPowerThresholdViolated          interface       15 seconds
70 ddmRxPowerThresholdViolated          interface       15 seconds
71 webMgtServerErrorTrap                webmgt          15 seconds
72 multiChassisIpcVlanUp                mcm             15 seconds
73 multiChassisIpcVlanDown              mcm             15 seconds
74 multiChassisMisconfigurationFailure  mcm             15 seconds
75 multiChassisHelloIntervalConsisFailu mcm             15 seconds
76 multiChassisStpModeConsisFailure     mcm             15 seconds
77 multiChassisStpPathCostModeConsisFai mcm             15 seconds
78 multiChassisVflinkStatusConsisFailur mcm             15 seconds
79 multiChassisStpBlockingStatus        mcm             15 seconds
80 multiChassisLoopDetected             mcm             15 seconds
81 multiChassisHelloTimeout             mcm             15 seconds
82 multiChassisVflinkDown               mcm             15 seconds
83 multiChassisVFLMemberJoinFailure     mcm             15 seconds
84 alaDHLVlanMoveTrap                   vlan            15 seconds
85 alaDhcpClientAddressAddTrap          ip-helper       15 seconds
86 alaDhcpClientAddressExpiryTrap       ip-helper       15 seconds
87 alaDhcpClientAddressModifyTrap       ip-helper       15 seconds
88 vRtrIsisDatabaseOverload             isis            15 seconds
89 vRtrIsisManualAddressDrops           isis            15 seconds
90 vRtrIsisCorruptedLSPDetected         isis            15 seconds
91 vRtrIsisMaxSeqExceedAttempt          isis            15 seconds
92 vRtrIsisIDLenMismatch                isis            15 seconds
93 vRtrIsisMaxAreaAddrsMismatch         isis            15 seconds
94 vRtrIsisOwnLSPPurge                  isis            15 seconds
95 vRtrIsisSequenceNumberSkip           isis            15 seconds
96 vRtrIsisAutTypeFail                  isis            15 seconds
97 vRtrIsisAuthFail                     isis            15 seconds
98 vRtrIsisVersionSkew                  isis            15 seconds
99 vRtrIsisAreaMismatch                 isis            15 seconds
100 vRtrIsisRejectedAdjacency           isis            15 seconds
101 vRtrIsisLSPTooLargeToPropagate      isis            15 seconds
102 vRtrIsisOrigLSPBufSizeMismatch      isis            15 seconds
103 vRtrIsisProtoSuppMismatch           isis            15 seconds
104 vRtrIsisAdjacencyChange             isis            15 seconds
105 vRtrIsisCircIdExhausted             isis            15 seconds
```

```
106 vRtrIsisAdjRestartStatusChange      isis            15 seconds
107 mvrpVlanLimitReachedEvent           bridge          15 seconds
108 alaHAVlanClusterPeerMismatch        ha-vlan         15 seconds
109 alaHAVlanMCPeerMismatch             ha-vlan         15 seconds
110 alaHAVlanDynamicMAC                 ha-vlan         15 seconds
111 unpMcLagMacIgnored                  da-unp          15 seconds
112 unpMcLagConfigInconsistency         da-unp          15 seconds
113 multiChassisGroupConsisFailure      mcm             15 seconds
114 multiChassisTypeConsisFailure       mcm             15 seconds
115 alaPimNonBidirHello                 ipmr            15 seconds
116 dot1agCfmFaultAlarm                 bridge          15 seconds
117 alaSaaIPIterationCompleteTrap       system          15 seconds
118 alaSaaEthIterationCompleteTrap      system          15 seconds
119 alaSaaMacIterationCompleteTrap      system          15 seconds
120 virtualChassisStatusChange          vcm             15 seconds
121 virtualChassisRoleChange            vcm             15 seconds
122 virtualChassisVflStatusChange       vcm             15 seconds
123 virtualChassisVflMemberPortStatusCh vcm             15 seconds
124 virtualChassisVflMemberPortJoinFail vcm             15 seconds
125 lldpV2RemTablesChange               aip             15 seconds
126 vRtrLdpInstanceStateChange          mpls            15 seconds
127 evbFailedCdcpTlvTrap                evb             15 seconds
128 evbFailedEvbTlvTrap                 evb             15 seconds
129 evbUnknownVsiManagerTrap            evb             15 seconds
130 evbVdpAssocTlvTrap                  evb             15 seconds
131 evbCdcpLldpExpiredTrap              evb             15 seconds
132 evbTlvExpiredTrap                   evb             15 seconds
133 evbVdpKeepaliveExpiredTrap          evb             15 seconds
134 smgrServiceError                    svcmgr          15 seconds
135 smgrServiceHwError                  svcmgr          15 seconds
136 smgrSapError                        svcmgr          15 seconds
137 smgrSapHwError                      svcmgr          15 seconds
138 smgrSdpError                        svcmgr          15 seconds
139 smgrSdpHwError                      svcmgr          15 seconds
140 smgrSdpBindError                    svcmgr          15 seconds
141 smgrSdpBindHwError                  svcmgr          15 seconds
142 smgrGeneralError                    svcmgr          15 seconds
143 smgrStatusChange                    svcmgr          15 seconds
144 portViolationNotificationTrap       interface       15 seconds
145 multiChassisConsisFailureRecovered  mcm             15 seconds
146 alaSaaPacketLossTrap                system          15 seconds
147 alaSaaJitterThresholdYellowTrap     system          15 seconds
148 alaSaaRTTThresholdYellowTrap        system          15 seconds
149 alaSaaJitterThresholdRedTrap        system          15 seconds
150 alaSaaRTTThresholdRedTrap           system          15 seconds
151 chassisTrapsDuplicateMacCleared     chassis         15 seconds
152 alaFipsResourceThresholdReached     fips            15 seconds
153 virtualChassisUpgradeComplete       vcm             15 seconds
154 appFPSignatureMatchTrap             appfp           15 seconds
155 virtualChassisVflSpeedTypeChange    vcm             15 seconds
156 alaSIPSnoopingACLPreemptedBySOSCall qos             15 seconds
157 alaSIPSnoopingRTCPOverThreshold     sip-snooping    15 seconds
158 alaSIPSnoopingRTCPPktsLost          qos             15 seconds
159 alaSIPSnoopingSignallingLost        qos             15 seconds
160 alaSIPSnoopingCallRecordsFileMoved  sip-snooping    15 seconds
161 alaIPv6NeighborLimitExceeded        ip              15 seconds
162 alaIPv6NeighborVRFLimitExceeded     ip              15 seconds
163 alaIPv6InterfaceNeighborLimitExceed ip              15 seconds
164 alaDyingGaspTrap                    interface       15 seconds
165 alaDhcpSrvLeaseUtilizationThreshold dhcp-server     15 seconds
166 alaDHCPv6SrvLeaseUtilizationThresho dhcpv6-server   15 seconds
167 smgrServiceStatusChange             svcmgr          15 seconds
168 smgrSapStatusChange                 svcmgr          15 seconds
169 smgrSdpStatusChange                 svcmgr          15 seconds
170 smgrSdpBindStatusChange             svcmgr          15 seconds
171 alaPethPwrSupplyConflictTrap        module          15 seconds
172 alaPethPwrSupplyNotSupportedTrap    module          15 seconds
173 chasTrapsBPSLessAllocSysPwr         chassis         15 seconds
174 chasTrapsBPSStateChange             chassis         15 seconds
175 chasTrapsNiBPSFETStateChange        chassis         15 seconds
176 alaDhcpBindingDuplicateEntry        ip-helper       15 seconds
177 alaVCSPProtectionTrap               vcm             15 seconds
178 alaVCSPRecoveryTrap                 vcm             15 seconds
179 pethPsePortOnOffNotification        module          15 seconds
180 pethMainPowerUsageOnNotification    module          15 seconds
181 pethMainPowerUsageOffNotification   module          15 seconds
182 chasTrapsBPSFwUpgradeAlert          chassis         15 seconds
183 alaAppMonAppRecordFileCreated       app-mon         15 seconds
184 alaAppMonFlowRecordFileCreated      app-mon         15 seconds
185 alaDPIFlowRecordFileCreated         dpi             15 seconds
186 alaLbdStateChangeToShutdown         lbd             15 seconds
```

```
187 alaLbdStateChangeForClearViolationA lbd            15 seconds
188 alaLbdStateChangeForAutoRecovery    lbd            15 seconds
```

The OS6860/OS6900/OS10K supports the SNMPv1 and SNMPv2c community strings security standards. When a community string is carried over an incoming SNMP request, the community string must match up with a user account name as listed in the community string database on the switch. Otherwise, the SNMP request will not be processed by the SNMP agent in the switch.

The **show snmp community-map** command shows the local community strings database, including status, community string text, and user account name. For example:

```
-> show snmp community-map
Community mode : enabled

status        community string                 user name
--------+-------------------------------+--------------------------------
enabled      public                                      public
```

SNMPv3 authentication is accomplished between the switch and the SNMP management station through the use of a username and password identified via the SNMP station CLI syntax. The username and password are used by the SNMP management workstation along with an authentication algorithm, either SHA or MD5, to compute a hash value that is transmitted in the PDU. When the switch receives the PDU, it will verify the authentication and encryption for validation.

To display the encryption type, use the **show user** command:

```
-> show user snmptest
User name = snmptest,
  Password expiration     = None,
  Password allow to be modified date     = None,
  Account lockout     = None,
  Password bad attempts     = 0,
  Read Only for domains   = None,
  Read/Write for domains  = ,
  Read/Write for families = snmp ,
  Snmp allowed     = YES,
  Snmp authentication     = SHA,
  Snmp encryption     = DES
  Console-Only    = Disabled
```

18.2. **SNMP Security**

By default, the switch is set to **privacy all**, which means the switch accepts only authenticated and encrypted v3 Sets, Gets, and Get-Nexts.To verify the SNMP security setting, use the **show snmp security** command:

```
-> show snmp security
snmp security = no security
```

18.3. SNMP Statistics

The **show snmp statistics** command can be very useful in determining if the switch is sending any traps. If the switch is sending traps but the workstation is not receiving them, the workstation may have an issue (for example, Windows firewall) or the IP address is not configured correctly, or user id, etc, on the switch.

Each MIB object displayed in the **show snmp statistics** command output is listed with a counter value.

For example:

```
-> show snmp statistics
From RFC1907
  snmpInPkts                       = 101030
  snmpOutPkts                      = 101030
  snmpInBadVersions                = 0
  snmpInBadCommunityNames          = 0
  snmpInBadCommunityUses           = 0
  snmpInASNParseErrs               = 0
  snmpEnableAuthenTraps            = enabled(1),
  snmpSilentDrop                   = 0
  snmpProxyDrops                   = 0
  snmpInTooBigs                    = 0
  snmpInNoSuchNames                = 0
  snmpInBadValues                  = 0
  snmpInReadOnlys                  = 0
  snmpInGenErrs                    = 0
  snmpInTotalReqVars               = 1099809
  snmpInTotalSetVars               = 3769
  snmpInGetRequests                = 39837
  snmpInGetNexts                   = 30099
  snmpInSetRequests                = 3769
  snmpInGetResponses               = 0
  snmpInTraps                      = 0
  snmpOutTooBigs                   = 0
  snmpOutNoSuchNames               = 0
  snmpOutBadValues                 = 0
  snmpOutGenErrs                   = 0
  snmpOutGetRequests               = 0
  snmpOutGetNexts                  = 0
  snmpOutSetRequests               = 0
  snmpOutGetResponses              = 101030
  snmpOutTraps                     = 642775
From RFC2572
  snmpUnknownSecurityModels        = 0
  snmpInvalidMsgs                  = 0
  snmpUnknownPDUHandlers           = 0
From RFC2573
  snmpUnavailableContexts          = 0
  snmpUnknownContexts              = 0
From RFC2574
  usmStatsUnsupportedSecLevels     = 0
  usmStatsNotInTimeWindows         = 0
  usmStatsUnknownUserNames         = 0
  usmStatsUnknownEngineIDs         = 0
  usmStatsWrongDigests             = 0
  usmStatsDecryptionErrors         = 0
```

By default the switch assigns UDP Port 1162 for the SNMP traps to be sent to the SNMP network management station, but the NMS station might be listening to some other port for the traps. Make sure that the switch matches with the NMS's port setup using the **show snmp station** command.

For example:

```
-> show snmp station
ipAddress/udpPort                                   status    protocol user
---------------------------------------------------+---------+--------+-------
10.100.10.21/1162                                    enable    v2       public
```

The switch normally stores all traps sent out to the SNMP management stations. To list the last stored traps by using the **show snmp-trap replay-ip** command. This command lists the traps along with their sequence number. The sequence number is a record of the order in which the traps were previously sent out.
For example:

```
-> show snmp-trap replay-ip
ipAddress                                 oldest replay number
-----------------------------------------+--------------------
10.100.10.21                                       0
```

Debug Command List

Use the debug snmp data community command to verify the community string configured:

```
-> debug snmp data community
Community (mode 2, counter 1) map :

0 @0x10097f8c : status 1
   community (size 6, name (public))
        user (size 9, name (snmpwrite))
```

Use the debug snmp data user command to verify the snmp user configuration configured:

```
-> debug snmp data user
0 @0x1009ad28 : status ALU_SNMP_USER_CREATED
   name snmpwrite, authPriv NOAUTH
   ASA read-write (0xffffffff,0xffffffff) read-only (0x0,0x0)
   traficTicks 89/300, refresTicks 4/5
```

Use the **debug trap counts** command to verify the list of traps generated:

```
-> debug trap counts
+------------------+
|   Trap Manager   |
+------------------+
Trap   Rcv  Fwd  absorbed dropped
[ 0]     4    3     1       0
[ 1]     1    1     0       0
[ 2]     8    8     0       0
[ 3]     9    9     0       0
[ 4]     4    3     1       0
[ 5]     2    2     0       0
[ 8]     1    1     0       0
[24]    36   26    10       0
[25]     7    6     1       0
[27]     1    1     0       0
[120]     1    1     0       0
[121]     1    1     0       0
[183]   315  315     0       0

Total  390  377     13       0
```

19. Troubleshooting Power Over Ethernet

Power over Ethernet (PoE) feature allows PoE -capable/Powered Devices (PD) to be powered up (such as IP phones, WLAN Access Points, IP cameras). The OmniSwitch supports the IEEE 802.3af and 802.3at standards. IEEE 802.3af (PoE) standard supports up to 15.4W and IEEE 802.3at (PoE+) standard supports up to 25.5W.

Models supported: OS6860-P24, OS6860-P48, OS6860E-P24, OS6860E-P48

The supported power supply on the OS6860 and OS6860E devices are indicated, as below:
OS6860-P24 uses OS6860-BPPH (Modular 600-W AC PoE power supply)
OS6860-P48 uses OS6860-BPPX (Modular 920-W AC PoE power supply)
OS6860E-P24 uses OS6860-BPPH (Modular 600-W AC PoE power supply)

Summary of the commands in this chapter is listed here:

---

show powersupply
show lanpower slot <chassis/slot>
show lanpower slot <chassis/slot> capacitor-detection
show lanpower slot <chassis/slot> class-detection
show lanpower slot <chassis/slot> priority-disconnect
show lanpower slot <chassis/slot> usage-threshold

19.1. Troubleshooting PoE on OmniSwitch on OS6860 and OS6860E

Command to verify PoEstatus: show lanpower slot <chassis/slot>
Command to verify power supply: show powersupply

Below is an example:

```
-> show powersupply
             Total     PS
Chassis/PS   Power     Type     Status   Location
-----------+---------+--------+--------+-----------
 1/1         920       AC       UP       Internal
    Total   920
```

PoEP ower Status

Default behaviour: PoE is disabled by default

PoE status command:

```
-> show lanpower slot <chassis/slot>
```

Below is an example:

```
-> show lanpower slot 1/1
Port Maximum(mW) Actual Used(mW)   Status    Priority   On/Off   Class   Type
----+-----------+---------------+-----------+---------+--------+-------+----------
  1     30000            0       Searching      Low      ON        *
  2      30000            0       Searching      Low       ON         *
  3      30000            0       Searching      Low       ON         *
  4      30000            0       Searching      Low       ON         *
  5      30000            0       Searching      Low       ON         *
  6      30000            0       Searching      Low       ON         *
  7      30000            0       Searching      Low       ON         *
  8      30000            0       Searching      Low       ON         *
  9      30000            0       Searching      Low       ON         *
 …
```

ChassisId 1 Slot 1 Max Watts 780
780 Watts Total Power Budget Used
0 Watts Total Power Budget Available
1 Power Supply Available
BPS power: Not Available

Definition of terms:

"ChassisId 1 Slot 1 Max Watts" refers to the Maximum watts allocated to the corresponding chassis and slot.
"Watts Total Power Budget Used" refers to the Power Budget for the PoE ports.
"Watts Total Power Budget Available" refers to the Amount of power budget remaining that can be allocated for additional switch functions.
"Power Supply Available" refers to the number of Power Supply.
"BPS power:" refers to the availability of the Redundant Power Supply.

Additional PoE features/commands:

1. Capacitor-detection
This feature is disabled by default. It is enabled when there are legacy devices (such as IP phones) attached to the corresponding slot. Note that this feature is not compatible with IEEE specifications.

```
-> show lanpower slot <chassis/slot> capacitor-detection
```

Below is an example:

```
-> show lanpower slot 1/1 capacitor-detection
Capacitor Detection disabled on ChassisId 1 Slot 1
```

2. Class-detection
This feature is disabled by default. When class detection is enabled, attached devices will automatically be limited to their class power, regardless of port power configuration.

```
-> show lanpower slot <chassis/slot> class-detection
```

Below is an example:

```
-> show lanpower slot 1/1 class-detection
Class Detection disabled on ChassisId 1 Slot 1
```

3. Priority-disconnect
This feature is enabled by default. Priority disconnect is used by the system software in determining whether an incoming PD will be granted or denied power when there are too few watts remaining in the PoE power budget for an additional device.

```
-> show lanpower slot <chassis/slot> priority-disconnect
```

Below is an example:

```
-> show lanpower slot 1/1 priority-disconnect
Priority Disconnect enabled on ChassisId 1 Slot 1
```

4. Usage-threshold:
This feature is set at 99(%) aby default. The switch checks for a user-defined, slot-wide threshold for PoE power usage, in percent. When the usage threshold is reached or exceeded, a notification is sent to the user.

```
-> show lanpower slot <chassis/slot> usage-threshold
```

Below is an example:

```
-> show lanpower slot 1/1 usage-threshold
Usage Threshold 99% on ChassisId 1 Slot 1
```

5. Power Priority:
The default power priority is Low.
• Low. This default value is used for port(s) that have low-priority devices attached. In the event of a power management issue, inline power to low-priority is interrupted first (i.e., before critical and high priority).
• High. This value is used for port(s) that have important, but not mission-critical, devices attached. If others in the chassis have been configured as critical, inline power to high-priority is given second priority.
• Critical. This value is used for port(s) that have mission-critical devices attached, and therefore require top (i.e., critical) priority. In the event of a power management issue, inline power to critical is maintained as long as possible.

Below is an example:

```
-> show lanpower slot 1/1
Port Maximum(mW) Actual Used(mW)   Status    Priority   On/Off   Class   Type
----+-----------+---------------+-----------+---------+--------+-------+----------
  1     30000           0       Searching      Low      ON        *
  2     30000           0       Searching      Low      ON        *
  3     30000           0       Searching      Low      ON        *
….
```

20. Troubleshooting Ethernet Ring Protection (ERP)

Ethernet Ring Protection (ERP) is a feature/algorithm that provides loop-free topology with redundancy and scalbility. Loop prevention is carried out throughout the links, with one of the links blocked. Implementation of ERP is based on the Recommendation ITU-T G.8032/Y.1344 standard.

ERP operates over standard Ethernet interfaces that are physically connected in a ring topology. In an Ethernet ring, each node is connected to two adjacent nodes using two independent links called ring links. A ring link is bound by two adjacent nodes on ports called ring ports. The ring nodes support standard FDB (Filtering database) MAC learning, forwarding, flush behavior, and port blocking and unblocking mechanisms.

Any failure along the ring triggers a R-APS(SF) (R-APS signal fail) message along both directions from the nodes adjacent to the failed link after these nodes have blocked the port facing the failed link. On obtaining this message, RPL owner unblocks the RPL port.

During the recovery phase when the failed link gets restored the nodes adjacent to the restored link send R-APS(NR) (R-APS no request) messages. On obtaining this message, the RPL owner block the RPL port and then sends a R-APS(NR,RB) (R-APS no request, root blocked) message. This will cause all other nodes other than RPL owner in the ring to unblock all the blocked ports.

A ring operates in one of two modes:
Idle mode is the normal operation when all links up and RPL is blocked and *Protection mode* occurs when protection switching is activated. A ring failure will trigger the RPL into a forwarding state).

By default, Spanning Tree will not operate on the ERP ring ports. When the port remains an ERP port it will not control the blocking/forwarding behavior of the port. Spanning Tree will be active on all other switch ports and will determine the blocking or forwarding state of VLANs configured on those ports. The switch can be configured for per-VLAN (1x1) mode or Flat mode.

Models supported: OS6860, OS6860E, OS6900 and OS10K

Summary of the commands in this chapter is listed here:

```
show erp
show erp statistics
show erp ring <ringid>
show erp statistics ring <ringId>
show erp port <chassis/slot/port>
debug qos internal "slot 1 list 1 verbose" | grep ERP
```

20.1. Troubleshooting ERP on OmniSwitch

ERP is disabled by default.
Below is the command to verify the overall ERP status:

```
-> show erp
```

Below is an example (default):

```
-> show erp
Legends: WTR - Wait To Restore
         MEG - Maintenance Entity Group

  Ring      Ring    Ring     Ring    Serv  WTR  Guard  MEG     Ring      Ring
   ID       Port1   Port2   Status   VLAN Timer Timer Level    State     Node
                                          (min) (csec)
----------+-------+------+---------+-----+-----+-----+-----+-----------+--------

Total number of rings configured = 0
```

Below is an example (when ERP is configured and running):

```
-> show erp
Legends: WTR - Wait To Restore
         MEG - Maintenance Entity Group

  Ring      Ring    Ring     Ring    Serv  WTR  Guard  MEG     Ring      Ring
   ID       Port1   Port2   Status   VLAN Timer Timer Level    State     Node
                                          (min) (csec)
----------+-------+------+---------+-----+-----+-----+-----+-----------+--------
         1   1/1/1  1/1/2   enabled  1001     5    50     1  protection      rpl

Total number of rings configured = 1
```

Below is the command to view the ERP statistics:

```
-> show erp statistics
```

Below is an example (default):

```
-> show erp statistics
                    Signal_Fail_PDUs      No_Request_PDUs    No_Req_Block_PDUs Invalid
Ring        Port   Sent   Recv   Drop    Sent   Recv  Drop   Sent   Recv   Drop  PDU rx
----------+------+------+------+------+------+------+------+------+------+------+-------
```

Below is an example (when ERP is configured and running):

```
-> show erp statistics
                    Signal_Fail_PDUs      No_Request_PDUs    No_Req_Block_PDUs Invalid
Ring        Port   Sent   Recv   Drop    Sent   Recv  Drop   Sent   Recv   Drop  PDU rx
----------+------+------+------+------+------+------+------+------+------+------+-------
         1  1/1/1     12      9      0      4     54      0      3   1234      0      0
         1  1/1/2      0      0      0      4     51      0      0   1234      0      0
```

Below is the command to view the status of a particular ring:

```
-> show erp ring <ringid>
```

Below is an example (when ERP is configured and running):

```
-> show erp ring 1
Ring Id                 : 1,
Ring Type               : Normal Ring,
Ring Port1              : 1/1/1,
Ring Port2              : 1/1/2,
Ring Status             : enabled,
Service VLAN            : 1001,
Revertive Mode          : enable,
WTR Timer (min)         : 5,
Guard Timer (centi-sec) : 50,
Virtual Channel         : enable,
MEG Level               : 1,
Ring State              : idle,
Active ERP version      : Ver 2,
Ring Node Type          : rpl,
RPL Port                : 1/1/1,
Last State Change       : 00h:36m:35.00s
```

ERP Versions and Parameters

There are two types of ERP versions supported, as below:

ERPv1 supports a single-ring topology with features of loop prevention and supports standard FDB (Filtering database) MAC learning, forwarding, flush behavior, and port blocking, and unblocking mechanisms.

ERPv2 supports multi-ring and ladder topologies that contain interconnection nodes, interconnected shared links, master rings and sub-rings. Multiple ERP instances are supported per physical ring, in addition to features on ERPv1.

Here is the list of the common terminologies/parameters used:

Automatic Protection Switching (APS) or Ring APS (R-APS), is a protocol used to coordinate protection and recovery switching mechanisms over the Ethernet ring.

Ring APS (Automatic Protection Switching) Messages are protocol messages defined in Y.1731 and G.8032 that determine the status of the ring.

Ring Protection Link (RPL) is a link blocked to avoid forming a loop in the ring.

Ring Blocked (RB) is a blocked RPL, blocked under normal conditions.

Signal Failure (SF) is a message sent on the ring to inform other ring nodes of the failure condition, when a link or port failure is detected.

Remote Maintenance End Point identifier (RMEPID) is identifier used to identify the endpoint.

Link Monitoring is the monitoring of links using standard ETH (Ethernet Layer Network) CC OAM messages. Note that for improved convergence times, this implementation also uses Ethernet link up and link down events.

Signal Fail (SF) is the status declared when a failed link or node is detected.

No Request (NR) is the status declared when there are no outstanding conditions (for example, SF) on the node.

ERP Service VLAN is the Ring-wide VLAN used exclusively for transmission of messages, including R-APS messages for Ethernet Ring Protection.

ERP Protected VLAN is the VLAN that is added to the ERP ring. ERP determines the forwarding state of protected VLANs.

Filtering Database (FDB) is a database that stores filtered data according to the R-APS messages recieved. This database also maintains an association table that identifies the master rings for a given sub-ring.

Blocked Port Reference (BPR) is a port that identifies the ring port ("0" for interconnection node or sub-ring, "1" for master ring) that is blocked. The BPR status is used in all R-APS messages.

Continuity Check Messages (CCM) are messages required to monitor the ring-port connectivity across the L2 network, when an Ethernet ring contains no ERP capable nodes.

Management Entity Group (MEG) are switches given with priority as Management Entity Group Level (MEL).

Not Reachable (NR) and Signal Failure (SF) are status messages that can be sent as part of the R-APS messages.

Wait To Restore (WTR) is a timer used by the RPL to verify stability of the Etherenet ring.

Guard Timer (GT) is used to prevent the ring nodes from receiving outdated R-APS messages that are no longer relevant. A ring node initiates the guard timer when the failed link recovers.

R-APS Messages

R-APS messages are continuously transmitted wherein the first 3 messages are transmitted simultaneously to ensure fast protection switching (if one or two R-APS messages are corrupted); and after that they are transmitted periodically with an interval of 5 seconds. Here are the types of messages:

R-APS (Signal Fail) message is continuously transmitted by the node that detects SF (Signal Fail) Condition until the condition persists and informs other nodes about the condition.

R-APS (No-Request, RPL Blocked) message is continuously transmitted by the RPL node to indicate the other nodes that there is no failure in the ring and RPL port is blocked.

R-APS (No-Request) message is continuously transmitted by the non RPL node that detects the clearing of SF (Signal Fail) until the reception of R-APS (NR, RB) from RPL node after WTR expiry.

R-APS (Event) message is transmitted as a single burst of 3 R-APS messages and is not continuously repeated beyond this burst. The transmission of this R-APS message is done in parallel to other R-APS messages. Flush messages are R-APS “event” messages transmitted using sub-code field.

State machine of ERP ring

Each ERP enabled ring can be in one of the three states, namely, *IDLE, PROTECTION and PENDING*.
At initialization, RPL node blocks its RPL port and unblocks its non RPL port and transmits R-APS (NR, RB). Also, non RPL nodes block one ring port and unblock other ring port. All the ERP nodes then go to the PENDING state. Then on reception of R-APS (NR, RB) from RPL node, all other non-rpl nodes unblock their blocked ring ports and all the ERP nodes then go to the IDLE state. So finally in *IDLE state* all the non RPL ports are in forwarding state and RPL port is in blocking state.

When the ring port (RPL or non RPL) of any ERP node goes down, then the EVENT "local SF" is detected by the node and R-APS (SF) is transmitted immediately from the other ring port. If the ring port (RPL or non RPL) that goes down is already blocked, then the Event "local SF" is detected by the node and R-APS (SF, DNF) is transmitted immediately from the other ring port. The node then unblocks its blocked port and blocks the down port. It is important to block the port which is going down so that when the port comes up it will be in the blocking state to avoid any loop. The node then flushes the FDB for both the ring ports and goes to the PROTECTION state. All other nodes receiving the R-APS (SF) PDU unblock their blocked port (RPL port at RPL node) and then go to the *PROTECTION state* and flush the FDB for their ring ports. If the node whose ring port goes down is RPL node then there will be no drop in the traffic and if the node is non ERP node then the traffic is switched through the protection path.

Similarly, if any node goes down then the nodes connected to that node detects the EVENT "local SF". If the node that goes down is non RPL, then the RPL port goes to the forwarding state on reception of R-APS (SF) PDU and then traffic is switched through it. And if RPL node goes down, then all the other nodes go to the PROTECTION state but there will be no drop in the traffic.

If the down ring port comes up for a node in *PROTECTION state*, then that node detects the EVENT "local Clear SF". The node starts the guard timer and then transmits R-APS (NR) if the node is non RPL or transmits R-APS (NR, RB) if the node is RPL. While the guard timer is running, the node will not process any in-coming R-APS PDU which allows us to ignore the out-dated R-APS PDUS that might be flowing in the network. If the ring node receiving R-APS (NR) message is having its ring ports block, then it compares the remote node ID information with its own node ID. If the remote node ID is higher than its own node ID then unblock its ring ports.

On reception of R-APS (NR) by RPL node and if the revertive mode is enabled, it will trigger the WTR timer and all the ERP nodes then go to the *PENDING state*. After the WTR expiry it blocks its RPL port and unblocks its non RPL port. It then transmits R-APS (NR, RB) PDU and flushes the FDB for ring ports. All the ERP nodes then go to the IDLE state. On reception of R-APS (NR, RB) by non RPL nodes, they unblock their ring ports and stop transmitting the R-APS PDU. They also flush the FDB for their ring ports.

Additional Commands

Below is the command to view the statistics for a particular ring:

```
-> show erp statistics ring <ringId>
```

Below is an example:

```
-> show erp statistics ring 1
Legends: R-APS - Ring Automatic Protection Switching
         RPL   - Ring Protection Link

Ring-Id : 1
  Ring Port : 1/1/1
    Signal Fail PDUs
      Sent : 12,
      Recv : 9,
      Drop : 0
    No Request PDUs
      Sent : 4,
      Recv : 54,
      Drop : 0
    No Request RPL Block PDUs
      Sent : 3,
      Recv : 1170,
      Drop : 0
    Invalid R-APS PDUs
      Recv : 0

  Ring Port : 1/1/2
    Signal Fail PDUs
      Sent : 0,
      Recv : 0,
      Drop : 0
    No Request PDUs
```

```
      Sent : 4,
      Recv : 51,
      Drop : 0
    No Request RPL Block PDUs
      Sent : 0,
      Recv : 1170,
      Drop : 0
    Invalid R-APS PDUs
      Recv : 0
```

Below is the command to view the status for a particular port:

```
-> show erp port <chassis/slot/port>
```

Below is an example:

```
->  show erp port 1/1/1
Ring-Id : 1
  Ring Port Status     : forwarding,
  Ring Port Type       : rpl,
  Ethoam Event         : disabled,
  Remote-endpoint Id   : none
```

Troubleshooting Commands

Commands will be provided accordingly from Support/Engineering, on a "case-to-case" basis.

Debug command

```
-> debug qos internal "slot 1 list 1 verbose" | grep ERP
```

21. Troubleshooting Shortest Path Bridging (SPB)

Shortest Path Bridging (SPB) supports SPB MAC (SPB-M) as defined in the IEEE 802.1aq standard. SPB-M is defined for use in Provider Backbone Bridge (PBB) networks as specified in the IEEE 802.1ah standard.

SPB-M provides a mechanism to automatically define a shortest path tree (SPT) bridging configuration through a Layer 2 Ethernet network. SPB-M Ethernet services use this configuration to encapsulate and tunnel data through the PBB network.
Shortest Path Bridging (SPB) implements frame forwarding on the shortest path between any two bridges in an Ethernet network. The shortest path trees (SPTs) calculated by SPB provide the shortest and most efficient path to and from the intended destination. SPTs are formed along the direct, straight-line links between switches to make up an overall path through the topology that provides a robust, efficient direction for network traffic to travel.

The bridging methodology needed to allow each bridge to serve as its own root bridge is enforced through the use of SPB BVLANs. This type of VLAN does not learn customer MAC addresses or flood unknown unicast and multicast traffic.
In This Chapter

- "Troubleshooting SPB on OmniSwitch OS6900/OS10K/OS6860"
- "SPB debug information"
- "Bshell Troubleshooting"

Summary of the commands in this chapter is listed here:

```
show service spb
show service isid
show service access
show service spb <service ID> ports
show service spb <service ID>  sap port <chassis/slot/port:sap id>
show service spb <service ID>  debug-info
show service spb <service ID>  counters
show service l2profile
show spb isis services
show spb isis nodes
show spb isis adjacency detail
d chg source_vp
d chg source_vp
```

21.1. Troubleshooting SPB on OmniSwitch OS6900/OS10K/OS6860 series

The **spb** parameter is used to display information about SPB services. SAP count displays the number of Service Access Points associated with this SPB service. Mcast mode provide the multicast replication mode (**Headend** or **Tandem**) for the service.

For example:

```
-> show service spb
Legend: * denotes a dynamic object
SPB Service Info
```

```
  SystemId : e8e7.32b3.4ccd,   SrcId : 0x34ccd,    SystemName : OS6860
                          SAP     Bind                      MCast
ServiceId   Adm  Oper Stats Count   Count   Isid      BVlan Mode     (T/R)
-----------+----+----+-----+-------+-------+---------+-----+--------------
2000        Up   Up    N    2       1       12000     4003  Headend  (0/0)
3000        Up   Up    N    1       1       13000     4003  Headend  (0/0)
Total Services: 2
```

The service ID is a unique number that identifies a specific SPB service. Information associated with the service ID is displayed.

```
-> show service spb 3000
SPB Service Detailed Info
  Service Id       : 3000,                 Description      : ,
  ISID             : 13000,                BVlan            : 4003,
  Multicast-Mode   : Headend,             TX/Rx Bits       : 0/0,
  Admin Status     : Up,                   Oper Status      : Up,
  Stats Status     : No,                   Vlan Translation : No,
  Service Type     : SPB,                  Allocation Type  : Static,
  MTU              : 9194,                 Def Mesh VC Id   : 3000,
  SAP Count        : 1,                    SDP Bind Count   : 1,
  Ingress Pkts     : 0,                    Ingress Bytes    : 0,
  Egress Pkts      : 0,                    Egress Bytes     : 0,
  Mgmt Change      : 03/07/2014 06:23:28, Status Change    : 03/07/2014 06:23:28

-> show service isid 13000
SPB Service Detailed Info
  Service Id       : 3000,                 Description      : ,
  ISID             : 13000,                BVlan            : 4003,
  Multicast-Mode   : Headend, TX/Rx Bits       : 0/0,
  Admin Status     : Up,                   Oper Status      : Up,
  Stats Status     : No,                   Vlan Translation : No,
  Service Type     : SPB,                  Allocation Type  : Static,
  MTU              : 9194,                 Def Mesh VC Id   : 3000,
  SAP Count        : 1,                    SDP Bind Count   : 1,
  Ingress Pkts     : 0,                    Ingress Bytes    : 0,
  Egress Pkts      : 0,                    Egress Bytes     : 0,
  Mgmt Change      : 03/07/2014 06:23:28, Status Change    : 03/07/2014 06:23:28
```

To display the access (customer-facing) port configuration for the bridge. By default all service access ports are displayed if a port or link aggregate number is not specified:

```
-> show service access
Port       Link  SAP     SAP     Vlan
Id         Status Type    Count   Xlation L2Profile                        Description
---------+------+-------+-------+-------+--------------------------------+----------------------
----------
2/1/10    Down   Manual  1       N       def-access-profile
2/1/23    Down   Manual  0       N       def-access-profile
2/1/26    Up     Manual  2       N       def-access-profile

Total Access Ports: 3
```

A SAP is a type of virtual port that is associated with a SPB service. To determine the SAP configuration for a specific service, use the **show service spb ports** command to view the virtual ports associated with a specific service.
For example:

```
-> show service spb 3000 ports
Legend: (*) dyn unicast object (+) remote mcast object (#) local mcast object
SPB Service 3000 Info
  Admin : Up,        Oper  : Up,     Stats      : N,         Mtu     : 9194,   VlanXlation : N,
  ISID  : 13000,     BVlan : 4003,   MCast-Mode : Headend,   Tx/Rx   : 0/0
                                       Sap Trusted:Priority/         Sap Description /
Identifier             Adm  Oper Stats Sdp SystemId:BVlan   Intf      Sdp SystemName
----------------------+----+----+-----+--------------------+--------+----------------------------
----
```

```
sap:2/1/26:0           Up   Up    N           Y:x           2/1/26      -
sdp:32787:3000*        Up   Up    Y    e8e7.32b3.365d:4003  2/1/27   OS6860

Total Ports: 2
```

To view configuration information for a specific SAP, use the **show service spb sap** command.
For example:

```
-> show service spb 3000 sap port 2/1/26:0
SAP Detailed Info
  SAP Id           : 2/1/26:0,             Description      : ,
  Admin Status     : Up,                  Oper Status      : Up,
  Stats Status     : No,                  Vlan Translation : No,
  Service Type     : SPB,                 Allocation Type  : Static,
  Trusted          : Yes,                 Priority         : 0,
  Ingress Pkts     : 0,                   Ingress Bytes    : 0,
  Egress Pkts      : 0,                   Egress Bytes     : 0,
  Mgmt Change      : 03/07/2014 05:47:38, Status Change    : 03/07/2014 05:50:50
```

21.2. SPB debug information

To display the debug information for the virtual ports associated with the SPB service. A virtual port represents a Service Access Point (SAP) or a Service Distribution Point (SDP) that is associated with the specified SPB service. In addition to the virtual port configuration, the command **show service spb debug-info** also provides the status and additional configuration information for the SPB service.

```
-> show service spb 3000 debug-info
Legend: (*) dyn unicast object (+) remote mcast object (#) local mcast object
SPB Service 3000 Debug Info
  Admin : Up,        Oper  : Up,     Stats      : N,         Mtu     : 9194,   VlanXlation : N,
  ISID  : 13000,     BVlan : 4003,   MCast-Mode : Headend,   Tx/Rx   : 0/0,
  VFI   : 2,         McIdx : 8190,   StatsHandle: 0

                                       Sap Trusted:Priority/         Sap Description /
Stats  /
Identifier             Adm  Oper Stats Sdp SystemId:BVlan   Intf      Sdp SystemName
VP     L2 McIdx
----------------------+----+----+-----+--------------------+---------+----------------------------
----+------+---------
sap:2/1/26:0           Up   Up    N           Y:x           2/1/26      -
3      0
sdp:32787:3000*        Up   Up    Y    e8e7.32b3.365d:4003  2/1/27   OS6860
4      1
Total Ports: 2
```

The command **show service spb counters** displays the traffic statistics for the specified SPB service and associated virtual ports. Use the **sap** parameter options with this command to display statistics for a specific SAP ID. A SAP ID is comprised of an access port (*slot*/*port* or *agg_id*) and an encapsulation value (**:0**, **:all**, **:***qtag*, or **:***outer_qtag.inner_qtag*) that is used to identify the type of customer traffic to map to the associated service.

```
-> show service spb 3000 counters
Legend: * denotes a dynamic object

Identifier              Ing Pkts   Ing Byte Count  Egr Pkts   Egr Byte Count
----------------------+----------+---------------+----------+----------------
sdp:32787:3000*         29         2722            545        47102
Total Ports: 1
```

The command **show service l2profile** display the Layer 2 profile configuration information for the bridge. This type of profile is applied to access (customer-facing) ports and specifies how to process Layer 2 protocol frames ingressing on such ports.

```
-> show service l2profile
```

```
Profile Name: def-access-profile
  STP        : tunnel,    802.1X     : drop,      802.3AD    : peer,      802.1AB    : drop,
  GVRP       : tunnel,    AMAP       : drop,      MVRP       : tunnel
```

The command **show spb isis services** displays the service instance identifier (I-SID) mapping for bridges participating in the SPB topology.This command provides a network-wide view of existing services to help verify that SPB services are correctly advertised and learned by ISIS-SPB.

```
-> show spb isis services
Legend: * indicates locally configured ISID
SPB ISIS Services Info:
                      System
    ISID      BVLAN   (Name : BMAC)                          MCAST(T/R)
------------+-------+---------------------------------------+-----------
*    12000     4003   OS6860              : e8:e7:32:b3:36:5d
*    12000     4003   OS6860              : e8:e7:32:b3:4c:cd
*    13000     4003   OS6860              : e8:e7:32:b3:36:5d
*    13000     4003   OS6860              : e8:e7:32:b3:4c:cd
ISIDs:     4
```

The command **show spb isis nodes** displays the discovered node-level parameter values for all of the ISIS-SPB switches participating in the topology. This command displays the system name, system ID, SPsource ID, and bridge priority parameter values for the bridges discovered within the ISIS-SPB topology.

```
-> show spb isis nodes
SPB ISIS Nodes:
 System Name          System Id      SourceID BridgePriority
--------------------+---------------+--------+---------------
 OS6860               e8e7.32b3.365d  0x3365d  32768 (0x8000)
 OS6860               e8e7.32b3.4ccd  0x34ccd  32768 (0x8000)
```

The command **show spb isis adjacency detail** displays information about the ISIS-SPB adjacencies created for the SPB bridge.

```
-> show spb isis adjacency detail
SPB ISIS Adjacency detail:
    SystemID: e8e7.32b3.365d :
      B-MAC         : e8:e7:32:b3:36:5d       , Hostname   : OS6860              ,
      Interface     : 2/1/27                  , Up Time    : Fri Mar  7 06:23:28 2014,
      State         : UP                      , DR Priority: 0                   ,
      Hold Time     : 21                      , Max Hold   : 27                  ,
      Adj Level     : L1                      , NLPIDs     : SPB                 ,
      ExtLocalCktId(YES): 1,
      Restart Support    : Disabled           ,
      Restart Status     : Not currently being helped,
      Restart Supressed  : Disabled
```

21.3. **Advanced Troubleshooting Scenarios**

Command MAC-Ping gives a way to check the connectivity with SPB domain. But how to find the path of one service thru SPB domain? Here is an example for it.

Purpose: find the path from device(MAC AAA) to Server (MAC SSS)
Step:

i. Login OmniVista, find MAC AAA lives on switch XXX and MAC SSS lives on switch YYY
ii. On switch XXX, do “show mac-learning” and find result

   SPB 4001:4001 e8:e7:32:d5:84:55 dynamic servicing sdp:32786:4001

   From this result, it shows isid 4001 bound to bvlan 4001 and it is using SDP 32786

iii. Do “show service sdp” and have the resul

   32786 00e0.b1e7.0bd3:4001 Up Up SPB

   It shows the MAC address 00e0.b1e7.0bd3 is the system-id of the remote switch at the other end of the SDP.

iv. Do" show spb isis spf bvlan 4001 bmac 00e0.b1e7.0bd3". Now it shows the path hop by hop of the shortest path SPB chose for this service flow.

```
SPB ISIS Path Details:
Path Hop Name        Path Hop BMAC
  --------------------+-------------------
XXX                e8:e7:32:cb:cf:03
ZZZ                e8:e7:32:cb:cd:35
YYY                e8:e7:32:d5:84:55
```

21.4. **bShell Troubleshooting**

Every SAP configured is considered as one virtual port. When a SAP is enabled, internally we will bring up the virtual port and configure the hardware with CML flags to 8 to do hardware learning for this virtual port.
For example:

```
BCM.0> d chg source_vp
SOURCE_VP.ipipe0[2]:
<VFI=1,TPID_ENABLE=1,SD_TAG_MODE=1,EXP_PVLAN_VID=1,EVEN_PARITY=1,ENTRY_TYPE=1,DVP=1,DISABLE_VLAN_
CHECKS=1,DEFAULT_VLAN_TAG=1,DEFAULT_VID=1,CML_FLAGS_NEW=8,CML_FLAGS_MOVE=8,>\
```

When a SAP is disabled, the virtual port still exists in hardware as well as in software. But in software we configure that virtual port down and configure hardware to set CML flag as 1 which is to drop the packet received in this virtual port.
For example:

```
BCM.0> d chg source_vp
SOURCE_VP.ipipe0[1]:
<VFI=1,TPID_ENABLE=1,SD_TAG_MODE=1,EXP_PVLAN_VID=1,EVEN_PARITY=1,ENTRY_TYPE=1,DVP=1,DISABLE_VLAN_
CHECKS=1,DEFAULT_VLAN_TAG=1,DEFAULT_VID=1,CML_FLAGS_NEW=1,CML_FLAGS_MOVE=1,>
```

22. Troubleshooting sFlow

Summary of the commands in this chapter is listed here:

show sflow sampler
debug sflow dump statistics

Short for "sampled flow", sFlow is an industry standard for packet export at Layer 2. An sFlow system consists of multiple devices performing two types of sampling: random sampling of packets or application layer operations, and time-based sampling of counters. The sampled packet/operation and counter information, referred to as flow samples and counter samples respectively, are sent as sFlow datagrams to a central server running software that analyzes and reports on network traffic; the sFlow collector. See sFlow.org consortium for sFlow protocol specifications.

Flow samples
Based on a defined sampling rate, an average of 1 out of N packets/operations is randomly sampled. This type of sampling does not provide a 100% accurate result, but it does provide a result with quantifiable accuracy.

Counter samples
A polling interval defines how often the network device sends interface counters. sFlow counter sampling is more efficient than SNMP polling when monitoring a large number of interfaces.

sFlow datagrams
The sampled data is sent as a UDP packet to the specified host and port. The official port number for sFlow is port 6343. The lack of reliability in the UDP transport mechanism does not significantly affect the accuracy of the measurements obtained from an sFlow agent. If counter samples are lost then new values will be sent when the next polling interval has passed. The loss of packet flow samples is a slight reduction in the effective sampling rate.
The UDP payload contains the sFlow datagram. Each datagram provides information about the sFlow version, the originating device's IP address, a sequence number, how many samples it contains and one or more flow and/or counter samples.

OS6900 and OS6860 allows sampling traffic at rate of 1:1 (meaning all packets are sampled):

```
6900> show sflow sampler

Instance  Interface  Receiver   Rate   Sample-Header-Size
-----------------------------------------------------------
   1      1/ 1          -          1          128
```

If sample rate is set to 1 and data rate is low, sFlow could get every packet, but if the data rate is high (e.g. 10G line rate), the sample rate will not be able to keep up and sampler will auto adjust to a higher sample rate. The configured sample rate is the lowest sample rate sFlow tries to achieve but is not guaranteed. The command "debug sflow show rate" will show the actual rate at the time the command is executed.

sFlow is not designed to sample at a rate of 1:1. The recommended sample rates are:
10mbps = 200
100mbps = 500
1,000mbps = 1000
10,000mbps = 2000

Packet sampling uses randomness in the sampling process to prevent synchronization with any periodic patterns in the traffic. While suggested packet sampling does not provide a 100% accurate result, it does provide a result with quantifiable accuracy. sflow.org provides examples and describes the basic techniques used to calculate results and quantify accuracy when processing packet sample data. If you use 3rd party software like InMon, this is already taken into account.

If the switch is experiencing congestion, the sample interval will increase.
This can be corrected by applying the following commands lines in the CLI which will reset the rate back to its originally configured rate.

```
sflow sampler 1 port 1/1 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/2 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/3 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/4 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/5 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/6 receiver 1 rate 5 sample-hdr-size 128
```

Configuration example:

```
sflow agent ip 192.168.10.14
sflow receiver 1 name sflow address 192.168.10.11 udp-port 6343 packet-size 1400 version 5
timeout 0
sflow sampler 1 port 1/1 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/2 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/3 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/4 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/5 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/6 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/35 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/36 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/37 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/38 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/39 receiver 1 rate 5 sample-hdr-size 128
sflow sampler 1 port 1/40 receiver 1 rate 5 sample-hdr-size 128
sflow poller 1 receiver 1 port 1/35 interval 5
sflow poller 1 receiver 1 port 1/36 interval 5
sflow poller 1 receiver 1 port 1/37 interval 5
sflow poller 1 receiver 1 port 1/38 interval 5
sflow poller 1 receiver 1 port 1/39 interval 5
sflow poller 1 receiver 1 port 1/40 interval 5
```

22.1. **sFLOW Debug**

CLI Command syntax:

```
DEBUG SFLOW DUMP STATISTICS
```

Dumps statistics on the datagrams transferred.

23. Troubleshooting Port Mirroring and Port Monitoring

Summary of the commands in this chapter is listed here:

show configuration snapshot pmm
show port-monitoring file
show port-monitoring status

23.1. Troubleshooting Port Mirroring

To verify the port mirroring configuration:

```
# show configuration snapshot pmm
! Port Mirroring:
port-mirroring 1 destination 2/1/14
port-mirroring 2 destination 2/1/17
port-mirroring 1 source 2/1/18 outport
port-mirroring 1 enable
port-mirroring 2 source 1/1/19 inport
port-mirroring 2 enable
```

The maximum number of supported port-mirroring sessions is two. Attempting to configure more than two will result in the error, “ERROR: exceeds the Max Number of Sessions”.

```
port-mirroring 3 source 1/1/13 destination 2/1/8
ERROR: Exceeds Max Number of Sessions
```

To remove the port-mirroring configuration issue the below commands:
# no port-mirroring 1
# no port-mirroring 2
# port-mirroring 1 source 1/1/1 destination 1/1/19 enable

If the destination is a 1/1/19 and it is part of a linkagg, the following error message is produced.

```
ERROR: Current Port State: LAG MEMBER -  , Failed to set Mirroring on port: 1/1/19
```

To correct this, choose another available port or remove the existing linkagg configuration from 1/1/19 etc.

To disable an existing port-mirroring session:

port-mirroring 2 disable
port-mi rroring 1 disable

```
# port-mirroring 2 source 2/1/14-15 destination 2/1/17
ERROR: Current Port State: FIXED - Invalid Property, Failed to set Mirrored on port: 2/1/14
```

This error message usually means the port 2/1/14 is already in-use as a destination port. To correct the configuration remove 2/1/14 as a destination port.

```
# port-mirroring 1 no source 2/1/18
ERROR: Session 1 is enabled. Cannot be modified.
```

*** disable port-mirroring 1 before you can remove any source port.

```
port-mirroring 1 disable
```

To remove destination ports, remove the entire port-mirroring session:

```
# show configuration snapshot pmm
! Port Mirroring:
port-mirroring 2 destination 2/1/17
port-mirroring 2 source 1/1/19 inport disable
port-mirroring 2 disable
# port-mirroring 1 source 2/1/18-20 destination 2/1/14 enable


# port-mirroring 1 source 2/1/18-20 destination 2/1/14-15 enable
                                                ^
ERROR: Invalid entry: "2/1/14-15"
```

Explanation: The destination can only be one port.

```
# port-mirroring 1 source 2/1/18-20 destination 2/1/14 enable
ERROR: Session 1 is enabled. Cannot be modified.

# port-mirroring 2 disable

# port-mirroring 2 source 1/1/19 inport enable

# port-mirroring 2 enable
```

Guidelines:

• A port mirroring and a port monitoring session can be configured on the same network interface module in an OmniSwitch OS10K, OS6900.
• A mirroring port can not be assigned to a tagged VLAN port.
• When a port is configured as a mirroring port, it's state is changed so that it does not belong to a VLAN. Inbound traffic to the mirroring port is dropped since it does not belong to a VLAN.
• Spanning tree is disabled by default on a mirroring port.
• Port mirroring is not supported on logical link aggregate ports. However, it is supported on individual ports that are members of a link aggregate.
• Execute the **port mirroring source destination** command to define the mirrored port and enable port mirroring status. Use the **port mirroring** command to enable the port mirroring session.
• Specify the *vlan_id* number of the mirroring port that is to remain **unblocked** when the command is executed. The **unblocked** VLAN becomes the default VLAN for the mirroring port. This VLAN handles the inbound traffic for the mirroring port. Spanning tree remains disabled on the unblocked VLAN.

23.2. Troubleshooting port monitoring

Note: Specify the entire path beginning with /flash

```
# port-monitoring 3 source 1/1/14 file "King.cap"
ERROR: Specify absolute path and no subdir eg: /flash/pmon.enc
```

The switch only supports 1 port-monitoring session.

```
# port-monitoring 4 source 1/1/14 file "/flash/portmon.cap"
ERROR: Exceeds Max Number of Sessions

# show configuration snapshot pmm
! Port Mirroring:
port-monitoring 3 source 1/1/14  file  /flash/portmon2.cap  size 1  timeout 0 bidirectional
capture-type brief
port-mirroring 1 destination 2/1/14
port-mirroring 2 destination 2/1/17
```

```
port-mirroring 1 source 2/1/18-20 bidirectional
port-mirroring 1 enable
port-mirroring 2 source 1/1/19 inport
port-mirroring 2 enable
# no port-mirroring 1
# port-monitoring 3 source 1/1/14 file /flash/portmon2.cap size 35
ERROR: Mirroring file size invalid
```

Note: The maximum file size for a port monitoring capture is is 35MB.

Miscellaneous Command Set:

```
# port-monitoring 1 source 1/1/14 file "/flash/portmon.cap" enable
ERROR: Exceeds Max Number of Sessions

# show configuration snapshot pmm
! Port Mirroring:
port-monitoring 3 source 1/1/14  file  /flash/portmon2.cap  size 1  timeout 0 bidirectional
capture-type brief
```

Note: Only one port monitoring session is permitted.

```
# ls pm3.cap
pm3.cap
# ls -l pm3.cap
-------r--    1 root     root       2319866 Jan  9 16:42 pm3.cap
# rm pm3.cap
rm: remove 'pm3.cap'? Y
# port-monitoring 3 source 1/1/19 file "/flash/pm3.cap" size 32 enable capture-type full

# ls -l pm3.cap
-------r--    1 root     root       2319933 Jan  9 16:46 pm3.cap
# ls -l pm3.cap
-------r--    1 root     root       2319933 Jan  9 16:46 pm3.cap

# port-monitoring 3 source 1/1/19 file "/flash/pm3.cap" size 32 disable capture-type brief

# ls -l *.cap
-------r--    1 root     root       2097107 Jan  9 16:53 pm3.cap
```

Note: The maximum file-size for port monitoring captures in brief is 2097107 Bytes
When used in brief mode, only the 1st 64-bytes of each packet are captured. Conversely in full mode the entire packet is captured.
To view the brief mac-addresses information, the following command can be issued:

```
show port-monitoring file


E8:E7:32:30:0A:69 | E8:E7:32:B8:AE:25 | II-8100| 81:00:00:06:08:00:45:00:00:BA

E8:E7:32:30:0A:69 | 00:00:02:05:00:8D | II-8100| 81:00:00:64:08:00:45:00:00:BB

E8:E7:32:30:0A:69 | E8:E7:32:B8:AE:25 | II-8100| 81:00:00:06:08:00:45:00:00:BB

data file is /flash/pm3.cap
```

To verify the status of the port monitoring session, the following command can be issued:

```
# show port-monitoring status

 Sess Mon.    Mon. Over  Oper.  Admin  Capt.   Max.   File
      Src     Dir  write Stat   Stat   Type    Size   Name
-----+-------+----+-----+------+------+-------+------+----------------------
  3.  1/1/19  Bi    ON    OFF   OFF    Full   2048K  /flash/pm3.cap
```

24. Troubleshooting IPV6

Summary of the commands in this chapter is listed here:

show ipv6 interface
show ipv6 routes
show ipv6 router database
show ipv6 traffic
show ipv6 route-pref
show ipv6 dhcp relay

24.1. IPv6 Routing

An IPv6 address can be configured on the switch for either a VLAN or a tunnel. Using the command show ipv6 interface verifies the IPv6 interface status.

```
-> show ipv6 interface
Name                             IPv6 Address/Prefix Length                        Status   Device
--------------------------------+-------------------------------------------------+--------+------------
v6if-v200                        2001:db8:4100:1000::/64                           Inactive VLAN 200
                                 2001:db8:4100:1000::40/64
                                 fe80::eae7:32ff:fed7:190d/64
tunnel_6to4                                                                        Disabled 6to4 Tunnel
loopback                         ::1/128                                           Active   Loopback

-> show ipv6 interface
Name                             IPv6 Address/Prefix Length                        Status   Device
--------------------------------+-------------------------------------------------+--------+------------
tunnel_6to4                                                                        Disabled 6to4 Tunnel
v6if-tunnel-137                  2100:db8:4132:4000::/64                           Active   Tunnel 1
                                 2100:db8:4132:4000:eae7:32ff:fed7:190d/64
                                 fe80::eae7:32ff:fed7:190d/64
loopback                         ::1/128                                           Active   Loopback
```

To view the IPv6 routing table, the below command can be used:

```
-> show ipv6 routes
Legend: Flags: U=Up, G=Gateway, H=Host, S=Static, C=Cloneable, B=Discard, E=ECMP

Total 2 routes

Destination/Prefix                                Gateway Address                         Interface
Age     Protocol Flag
----+----------+--------------------------------+---------------------------+-------------------------+------
::1/128                                           ::1                                           loopback
01:20:32 LOCAL    UH
2001:db8:4100:1000::/64                           fe80::eae7:32ff:feae:7811                     v6if-v200
00:20:28 LOCAL    UC


-> show ipv6 router database
Legend: + indicates routes in-use

Total IPRM IPv6 routes: 2

  Destination/Prefix                                Gateway Address                               Interface
Protocol   Metric        Tag
---------------------------------------------------+---------------------------------------------+-------------
+ ::1/128                                           ::1                                           loopback
LOCAL          1          0
+ 2001:db8:4100:1000::/64                           fe80::eae7:32ff:feae:7811                     v6if-v200
LOCAL          1          0

Inactive Static Routes:
Vlan  Destination/Prefix                            Gateway Address                               Metric
Tag
-----+---------------------------------------------+---------------------------------------------+-------+-----
-----
```

```
> show ipv6 traffic
Message                  Current    Previous   Change
------------------------+----------+----------+----------
Packets received
  Total                        23         10         13
  Header errors                 0          0          0
  Too big                       0          0          0
  No route                      0          0          0
  Address errors                0          0          0
  Unknown protocol              0          0          0
  Truncated packets             0          0          0
  Local discards                0          0          0
  Delivered to users           13          5          8
  Reassembly needed             0          0          0
  Reassembly failed             0          0          0
  Multicast packets             8          4          4
Packets sent
  Forwarded                     0          0          0
  Generated                    34         23         11
  Local discards                2          2          0
  Fragmented                    0          0          0
  Fragmentation failed          0          0          0
  Fragments generated           0          0          0
  Multicast packets            44         36          8

sno-lab-r1-6860> show ipv6 route-pref
  Protocol    Route Preference Value
------------+------------------------
  Local                 1
  Static                2
  OSPF                110
  ISISL1              115
  ISISL2              118
  RIP                 120
  EBGP                190
  IBGP                200
```

The show ipv6 traffic command gives switch-wide statistics for IPv6 traffic. The value for “No Route Discards” should be similar to the “icmp stats destination unreachable” number, and both values should be increasing at a similar rate. “No Route Discards” on a network is a normal occurrence, but the values should be increasing at a similar rate.

The route preference value of IPv4 is different than IPv6.

24.2. Troubleshooting DHCPv6 Relay

The DHCPv6 Relay on the OmniSwitch processes and forwards all DHCPv6 messages between clients and the configured DHCPv6 relay agent as a unicast packet.
A maximum of five unicast or link-scoped multicast relay destinations can be configured for each interface on which DHCPv6 Relay is enabled. The DHCPv6 relay for the interface will be automatically disabled when all the relay destinations configured for that interface are removed.

```
> show ipv6 dhcp relay
DHCPv6 Relay: Enabled
```

When the relay interface and the relay destination are configured the output is below:

```
> show ipv6 dhcp relay
DHCPv6 Relay: Enabled

Interface Relay Destination(s) Status
---------------------------+------------------------------------------+--------
```

```
vlan-41 ff02::1:2 Enabled
vlan-103 2001:dbc8:8003::17 Disabled
2001:dbc8:8004::99
vlan-200 fe80::cd0:deff:fe28:1ca5 vlan-201 Enabled
tunnel-2 2001:dbc8:a23::ea77 Enabled
```

24.3. **Troubleshooting a 6to4 Tunnel**

Using command show ipv6 interface, verify the tunnel interface is configured correctly.

```
-> show ipv6 interface
Name                             IPv6 Address/Prefix Length                        Status
Device
--------------------------------+-------------------------------------------------+--------+-----
-------
v6if-v200                        2001:db8:4100:1000::/64                           Inactive VLAN
200
                                 2001:db8:4100:1000::40/64
                                 fe80::eae7:32ff:fed7:190d/64
tunnel_6to4                                                                        Disabled 6to4
Tunnel
loopback                         ::1/128                                           Active
Loopback
```

The 6to4 relay router will advertise a route to 2002::/16 on its IPv6 router interface.

```
-> show ipv6 routes
Legend:Flags:U = Up, G = Gateway, H = Host, S = Static, C = Cloneable, D = Dynamic,
M = Modified, R = Unreachable, X = Externally resolved, B = Discard,
L = Link-layer, 1 = Protocol specific, 2 = Protocol specific
Destination Prefix Gateway Address Interface Age Protocol Flags
-------------------+----------------+--------+------------------+------------+---------+-----
::/0 2002:d468:8a89::137 v6if-6to4-137 18h 47m 26s Static UGS
137:35:35::/64 fe80::2d0:95ff:fe12:f470 v6if-tunnel-137 18h 51m 55s Local UC
195:35::/64 fe80::2d0:95ff:fe12:f470 v6if-to-eagle 18h 51m 55s Local UC
212:95:5::/64 fe80::2d0:95ff:fe12:f470 smbif-5 18h 51m 55s Local UC
2002::/16 2002:d423:2323::35 v6if-6to4-137 18h 51m 55s Other U
```