Cisco Nexus 1000V for KVM Troubleshooting Guide

First Published: 2015-12-17
Last Updated: 2015-12-18

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco Nexus 1000V for KVM Troubleshooting Guide

CONTENTS

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

Overview

This chapter introduces the basic concepts, methodology, and general troubleshooting guidelines for problems that might occur when you configure and use the Cisco Nexus 1000V.

Overview of the Troubleshooting Process

To troubleshoot your network, follow these steps:

Step 1 Gather information that defines the specific symptoms.

Step 2 Identify all potential problems that could be causing the symptoms.

Step 3 Systematically eliminate each potential problem (from most likely to least likely) until the symptoms disappear.

Overview of Best Practices

Best practices are the recommended steps that you should take to ensure the proper operation of your network. We recommend the following general best practices for most networks:

- Maintain a consistent Cisco Nexus 1000V release across all network devices.
- Refer to the release notes for your Cisco Nexus 1000V release for the latest features, limitations, and caveats.
- Enable system message logging. See Overview of Symptoms, page 1-3.
- Verify and troubleshoot any new configuration changes after implementing the change.

Troubleshooting Basics

This section introduces questions to ask when troubleshooting a problem with the Cisco Nexus 1000V or connected devices. Use the answers to these questions to identify the scope of the problem and to plan a course of action.

Troubleshooting Guidelines

By answering the questions in the following subsections, you can determine the paths that you need to follow and the components that you should investigate further.

Answer the following questions to determine the status of your installation:

- Is this a newly installed system or an existing installation? (It could be a new host, switch, or VLAN).
- Has the host ever been able to see the network?
- Are you trying to solve an existing application problem (too slow, too high latency, excessively long response time) or did the problem show up recently?
- What changed in the configuration or in the overall infrastructure immediately before the applications started to have problems?

To discover a network problem, follow these steps:

Step 1 Gather information about the problems in your system. See Gathering Information, page 1-2.

Step 2 Verify the Layer 2 connectivity. See Verifying Layer 2 Connectivity, page 1-3.

Step 3 Verify the configuration for your end devices (storage subsystems and servers).

Step 4 Verify end-to-end connectivity. See Verifying Layer 3 Connectivity, page 1-3.

Gathering Information

This section highlights the tools that are commonly used to troubleshoot problems within your network. These tools are a subset of what you might use to troubleshoot your specific problem.

Each chapter in this guide includes additional tools and commands that are specific to the symptoms and possible problems covered in that chapter.

You should also have an accurate topology of your network to help isolate problem areas.

Enter the following commands and examine the outputs:

- **show module**
- **show version**
- **show running-config**
- **show logging log**
- **show interfaces brief**
- **show vlan**
- **show accounting log**
- **show tech-support svs**

Note To enter commands with the **internal** keyword, you must log in with the network-admin role.

Verifying Ports

Answer the following questions to verify ports:

- Are you using the correct media copper or optical fiber type.
- Is the media broken or damaged?
- Are you checking a virtual Ethernet port? If so, enter the **show interface brief** command. The status should be up.
- Are you checking a physical Ethernet port? If so, you need to check it by looking at the server or by looking at an upstream switch.

Verifying Layer 2 Connectivity

Answer the following questions to verify layer 2 connectivity:

- Are the necessary interfaces in the same VLANs?
- Are all ports in a port channel configured the same for speed, duplex, and trunk mode?

Enter the **show vlan brief** command to check the status of a VLAN. The status should be up.

Enter the **show port-profile** command to check a port profile configuration.

Enter the **show interface brief** command to check the status of a virtual Ethernet port or a physical Ethernet port.

Verifying Layer 3 Connectivity

Answer the following questions to verify Layer 3 connectivity:

- Have you configured a default route?
- Are any IP access lists, filters, or route maps blocking route updates?

Use the **ping** or **trace** commands to verify connectivity. See the following topics for more information:

- Ping, page 2-1
- Traceroute, page 2-1

Overview of Symptoms

The symptom-based troubleshooting approach provides multiple ways to diagnose and resolve problems. By using multiple entry points with links to solutions, this guide serves users who might have identical problems that are perceived by different indicators. Search this guide in PDF form, use the index, or rely on the symptoms and diagnostics listed in each chapter as entry points to access necessary information in an efficient manner.

Using a given a set of observable symptoms on a network, it is important to be able to diagnose and correct software configuration issues and inoperable hardware components so that the problems are resolved with minimal disruption to the network. Those problems and corrective actions include the following:

- Identify key Cisco Nexus 1000V troubleshooting tools.

- Obtain and analyze protocol traces using SPAN or Ethanalyzer on the CLI.
- Identify or rule out physical port issues.
- Identify or rule out switch module issues.
- Diagnose and correct Layer 2 issues.
- Diagnose and correct Layer 3 issues.
- Obtain core dumps and other diagnostic data for use by Cisco TAC.
- Recover from switch upgrade failures.

System Messages

The system software sends the syslog (system) messages to the console (and, optionally, to a logging server on another system) during operation. Not all messages indicate a problem with your system. Some messages are purely informational, while others might help diagnose problems with links, internal hardware, or the system software.

System Message Text

Message-text is a text string that describes the condition. This portion of the message might contain detailed information about the event, including terminal port numbers, network addresses, or addresses that correspond to locations in the system memory address space. Because the information in these variable fields changes from message to message, it is represented here by short strings enclosed in square brackets. A decimal number, for example, is represented as [dec].

Syslog Server Implementation

The syslog facility allows the Cisco Nexus 1000V device to send a copy of the system message log to a host for more permanent storage. This process can be useful if the logs need to be examined over a long period of time or when the Cisco Nexus 1000V device is not accessible.

This example demonstrates how to configure a Cisco Nexus 1000V device to use the syslog facility on a Solaris platform. Although a Solaris host is being used, syslog configuration on all UNIX and Linux systems is very similar.

Syslog uses the concept of a facility to determine how it should be handled on the syslog server (the Solaris system in this example) and the message severity. Therefore, different message severities can be handled differently by the syslog server. They could be logged to different files or e-mailed to a particular user. Specifying a severity specifies that all messages of that level and greater severity (lower number) will be acted upon.

Note The Cisco Nexus 1000V messages should be logged to a different file from the standard syslog file so that they are not confused with other non-Cisco syslog messages. To prevent log messages from filling up the / file system, the log file should not be located on the / file system.
Syslog Client: switch1
Syslog Server: 172.22.36.211 (Solaris)

Syslog facility: local1
Syslog severity: notifications (level 5, the default)
File to log Cisco Nexus 1000V messages to: /var/adm/nxos_logs

To configure a syslog server, follow these steps:

Step 1 Configure the Cisco Nexus 1000V:

```
n1000v# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
n1000v(config)# logging server 192.0.2.1 6 facility local1
```

This example shows how to display the configuration:

```
n1000v# show logging server
Logging server: enabled
{192.0.2.1}
     server severity: notifications
     server facility: local1
```

Step 2 Configure the syslog server as follows:

a. Modify /etc/syslog.conf to handle local1 messages. For Solaris, there must be at least one tab between the facility severity and the action (/var/adm/nxos_logs).

```
#Below is for the NX-OS logging
local1.notice /var/adm/nxos_logs
```

b. Create the log file.

```
# touch /var/adm/nxos_logs
```

c. Restart the syslog.

```
# /etc/init.d/syslog stop
# /etc/init.d/syslog start
syslog service starting.
```

d. Verify the syslog started.

```
# ps -ef |grep syslogd
root 23508 1 0 11:01:41 ? 0:00 /usr/sbin/syslogd
```

Step 3 Test the syslog server by creating an event in the Cisco Nexus 1000V. In this case, port e1/2 was bounced and the following was listed on the syslog server. The IP address of the switch is listed in brackets.

```
# tail -f /var/adm/nxos_logs
Sep 17 11:07:41 [172.22.36.142.2.2] : 2004 Sep 17 11:17:29 pacific:
%PORT-5-IF_DOWN_INITIALIZING: %$VLAN 1%$ Interface e 1/2 is down (Initializing)
Sep 17 11:07:49 [172.22.36.142.2.2] : 2004 Sep 17 11:17:36 pacific: %PORT-5-IF_UP:
%$VLAN 1%$ Interface e 1/2 is up in mode access
Sep 17 11:07:51 [172.22.36.142.2.2] : 2004 Sep 17 11:17:39 pacific:
%VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from pts/0
(dhcp-171-71-49-125.cisco.com)
```

Troubleshooting with Logs

The Cisco Nexus 1000V generates many types of system messages on the switch and sends them to a syslog server. These messages can be viewed to determine what events caused a problem.

Viewing Logs

This example shows how to access and view logs in the Cisco Nexus 1000V:

```
switch# show logging ?

  <CR>
  >             Redirect it to a file
  >>            Redirect it to a file in append mode
  console       Show console logging configuration
  info          Show logging configuration
  internal      Logging internal information
  ip            IP configuration
  last          Show last few lines of logfile
  level         Show facility logging configuration
  logfile       Show contents of logfile
  module        Show module(linecard) logging configuration
  monitor       Show monitor logging configuration
  pending       Server address pending configuration
  pending-diff  Server address pending configuration diff
  server        Show server logging configuration
  session       Show logging session status
  status        Show logging status
  timestamp     Show logging timestamp configuration
  |             Pipe command output to filter
```

Example 1-1 shows an example of the **show logging** command output.

Example 1-1 show logging Command

```
switch# show logging server
Logging server: enabled
{192.0.1.1}
server severity: critical
server facility: user
```

Cisco Support Communities

For additional information, visit one of the following support communities:

- Cisco Support Community for Server Networking
- Cisco Communities: Nexus 1000V

Contacting Cisco Customer Support

If you cannot solve a problem after using the troubleshooting suggestions in this guide, contact a customer service representative for assistance. Before you call, have the following information ready:

- Version of the Cisco Nexus 1000V software that you are running
- Version of the Linux and OpenStack software that you are running
- Contact phone number
- Brief description of the problem
- Brief explanation of the steps that you have already taken to isolate and resolve the problem

If you purchased the product and support contract from Cisco, contact Cisco for support. Cisco provides Layer 1, Layer 2, and Layer 3 support.

For more information on steps to take before contacting customer support, see Gathering Information for Technical Support, page 16-1.

Tools Used in Troubleshooting

This chapter describes the troubleshooting tools available for the Cisco Nexus 1000V.

Commands

You use the CLI from a local console or remotely using a Telnet or Secure Shell (SSH) session. The command-line interface (CLI) provides a command structure similar to the Cisco NX-OS software, with context-sensitive help, **show** commands, multi-user support, and role-based access control.

Each feature has **show** commands that provide information about the feature configuration, status, and performance. Additionally, you can use the following commands for more information:

- **show system—Provides information on system-level components, including cores, errors, and exceptions. Use the show system error-id command to find details on error codes:**

```
n1000v# copy running-config startup-config
[########################################] 100%
2008 Jan 16 09:59:29 zoom %$ VDC-1 %$ %BOOTVAR-2-AUTOCOPY_FAILED: Autocopy of file
/bootflash/n1000-s1-dk9.4.0.0.837.bin.S8 to standby failed, error=0x401e0008

n1000v# show system error-id 0x401e0008
Error Facility: sysmgr
Error Description: request was aborted, standby disk may be full
```

Ping

The ping utility generates a series of *echo* packets to a destination across a TCP/IP internetwork. When the echo packets arrive at the destination, they are rerouted and sent back to the source. Using ping, you can verify connectivity and latency to a particular destination across an IP routed network.

The ping utility allows you to ping a port or end device. By specifying the IPv4 address, you can send a series of frames to a target destination. Once these frames reach the target, they are looped back to the source and a time stamp is taken. Ping helps you to verify the connectivity and latency to the destination.

Traceroute

Use traceroute to do the following:

- Trace the route followed by the data traffic.

- Compute inter-switch (hop-to-hop) latency.

Traceroute identifies the path taken on a hop-by-hop basis and includes a time stamp at each hop in both directions. You can use traceroute to test the connectivity of ports along the path between the generating switch and the switch closest to the destination.

Enter the **traceroute** command to access this feature.

If the destination cannot be reached, the path discovery starts, which traces the path up to the point of the failure.

Monitoring Processes and CPUs

The CLI has features that enable you to monitor switch processes and CPU status and utilization.

Identifying the Processes Running and Their States

Use the **show processes command** to identify the processes that are running and the status of each process. (See Example 2-1.) The command output includes the following:

- PID—Process ID.
- State—Process state.
- PC—Current program counter in hex format.
- Start_cnt—How many times a process has been started (or restarted).
- TTY—Terminal that controls the process. A "-" (hyphen) usually means a daemon that is not running on any particular TTY.
- Process—Name of the process.

Process states are as follows:

- D—Uninterruptible sleep (usually I/O).
- R—Runnable (on run queue).
- S—Sleeping.
- T—Traced or stopped.
- Z—Defunct (zombie) process.
- NR—Not-running.
- ER—Should be running but currently not-running.

Note The ER state typically designates a process that has been restarted too many times, which causes the system to classify it as faulty and disable it.

Example 2-1 show processes Command

```
n1000v# show processes

PID    State  PC        Start_cnt    TTY   Process
-----  -----  --------  -----------  ----  -------------
    1      S  41520eb8            1     -  init
    2      S         0            1     -  kthreadd
```

```
   3      S         0            1     -  migration/0
   4      S         0            1     -  ksoftirqd/0
   5      S         0            1     -  watchdog/0
   6      S         0            1     -  migration/1
   7      S         0            1     -  ksoftirqd/1
   8      S         0            1     -  watchdog/1
   9      S         0            1     -  events/0
  10      S         0            1     -  events/1
  11      S         0            1     -  khelper
  12      S         0            1     -  kblockd/0
  13      S         0            1     -  kblockd/1
  14      S         0            1     -  kacpid
  15      S         0            1     -  kacpi_notify
  16      S         0            1     -  kseriod
  17      S         0            1     -  ata/0
  18      S         0            1     -  ata/1
  19      S         0            1     -  ata_aux
  20      S         0            1     -  ksuspend_usbd
  21      S         0            1     -  khubd
  22      S         0            1     -  pdflush
  23      S         0            1     -  pdflush
  24      S         0            1     -  kswapd0
  25      S         0            1     -  aio/0
  26      S         0            1     -  aio/1
  27      S         0            1     -  nfsiod
  28      S         0            1     -  rpciod/0
  29      S         0            1     -  rpciod/1
  30      S         0            1     -  kirqd
 359      S         0            1     -  kjournald
 364      S         0            1     -  kjournald
 954      S         0            1     -  kjournald
 961      S         0            1     -  kjournald
1263      S  4151e5b6            1     -  portmap
1272      S  41520eb8            1     -  rpc.statd
1287      S         0            1     -  lockd
1288      S         0            1     -  nfsd
1289      S         0            1     -  nfsd
1290      S         0            1     -  nfsd
1291      S         0            1     -  nfsd
1292      S         0            1     -  nfsd
1293      S         0            1     -  nfsd
1294      S         0            1     -  nfsd
1295      S         0            1     -  nfsd
1300      S  41520eb8            1     -  rpc.mountd
1323      S  41520eb8            1     -  sysmgr
1675      S  41528053            1     -  httpd
1846      S         0            1     -  mping-thread
1847      S         0            1     -  mping-thread
1875      S         0            1     -  stun_kthread
1876      S         0            1     -  stun_arp_mts_kt
1877      S         0            1     -  stun_packets_re
1878      S         0            1     -  stun_send_packe
1933      S         0            1     -  redun_kthread
1934      S         0            1     -  redun_timer_kth
2269      S         0            1     -  sf_rdn_kthread
2286      S  41520eb8            1     -  xinetd
2287      S  41520eb8            1     -  tftpd
2288      R  4151e5ed            1     -  syslogd
2289      S  41520eb8            1     -  sdwrapd
2290      S  41520eb8            1     -  platform
2299      S         0            1     -  ls-notify-mts-t
2317      S  41520494            1     -  pfm_dummy
2319      S  41520494            1     -  vshd
2320      S  41520eb8            1     -  stun
```

```
2321      S  415c4642            1     -  smm
2322      S  41520eb8            1     -  redun_mgr
2323      S  41520eb8            1     -  psshelper
2324      S  41520eb8            1     -  lmgrd
2325      S  41520494            1     -  licmgr
2326      S  41520eb8            1     -  fs-daemon
2327      S  41520eb8            1     -  feature-mgr
2328      S  41520eb8            1     -  confcheck
2329      S  41520eb8            1     -  cdm
2330      S  41520eb8            1     -  capability
2331      S  41520eb8            1     -  psshelper_gsvc
2350      S  41520eb8            1     -  cisco
2351      S  41523f92            1     -  clis
2353      S  41520eb8            1     -  vem_mgr
2354      S  41523f92            1     -  port-profile
2357      S  41520eb8            1     -  xmlma
2358      S  41520ee7            1     -  vnm_pa_intf
2359      S  41520eb8            1     -  vmm
2360      S  41520eb8            1     -  vdc_mgr
2361      S  41520eb8            1     -  ttyd
2362      R  414f2c20            1     -  sysinfo
2363      S  41520eb8            1     -  sksd
2365      S  415277b3            1     -  res_mgr
2366      S  41520ee7            1     -  plugin
2367      S  415c4642            1     -  npacl
2368      S  41520eb8            1     -  mvsh
2369      S  41520eb8            1     -  mping_server
2370      S  41520eb8            1     -  module
2371      S  41523f92            1     -  fwm
2372      S  41520eb8            1     -  evms
2373      S  41520eb8            1     -  evmc
2374      S  41520eb8            1     -  core-dmon
2375      S  41520eb8            1     -  bootvar
2376      S  41520494            1     -  ascii-cfg
2377      S  41520494            1     -  securityd
2378      S  41523f92            1     -  cert_enroll
2379      S  41520eb8            1     -  aaa
2389      S  415c4642            1     -  l3vm
2390      S  415c4642            1     -  urib
2393      S  41520eb8            1     -  ExceptionLog
2394      S  41520eb8            1     -  ifmgr
2396      S  41520eb8            1     -  tcap
2415      S  41523f92            1     -  snmpd
2432      S  415c4642            1     -  adjmgr
2436      S  415c4642            1     -  u6rib
2461      S  41487a55            1     -  PMon
2467      S  41520eb8            1     -  aclmgr
2475      S  415c4642            1     -  arp
2476      S  414886c1            1     -  icmpv6
2480      S  415c4642            1     -  netstack
2552      S  b7f8757e            1     -  klogd
2571      S  41523f92            1     -  radius
2572      S  41520494            1     -  ip_dummy
2574      S  41520494            1     -  ipv6_dummy
2576      S  41523f92            1     -  ntp
2577      S  41520494            1     -  pktmgr_dummy
2578      S  41520494            1     -  tcpudp_dummy
2579      S  41523f92            1     -  dcos-xinetd
2581      S  41523f92            1     -  ntpd
2582      S  41523f92            1     -  cdp
2754      S  41523f92            1     -  ufdm
2755      S  41523f92            1     -  stp
2756      S  41523f92            1     -  seg_bd
2757      S  41523f92            1     -  sal
```

```
 2758      S  415c4642            1     -  rpm
 2759      S  41523f92            1     -  pltfm_config
 2760      S  41523f92            1     -  monitor
 2761      S  41520eb8            1     -  m2rib
 2762      S  41520eb8            1     -  ipqosmgr
 2763      S  415c4642            1     -  igmp
 2764      S  41523f92            1     -  eth_port_channel
 2765      S  41523f92            1     -  eth-port-sec
 2766      S  41520eb8            1     -  acllog
 2776      S  41520eb8            1     -  lacp
 2778      S  41523f92            1     -  vlan_mgr
 2798      S  41523f92            1     -  ethpm
 2844      S  41520eb8            1     -  msp
 2847      S  41523f92            1     -  vms
 2866      S  41523f92            1     -  vns_agent
 2867      S  41520eb8            1     -  vim
 2868      S  41523f92            1     -  nsmgr
 2869      S  41523f92            1     -  nfm
 2870      S  41523f92            1     -  httpmgr
 2871      S  41520eb8            1     -  cloud_agent
 2872      S  41520eb8            1     -  aclcomp
 2888      S  414f265e            1     -  ExtensibleApiEngine
 2890      S  414f265e            1     -  monitor_eae.sh
 2893      S  41520ee7            1     -  lua
 2903      S  414f265e            1     -  launch_apache.s
 2930      S  41520eb8            1     -  httpd
 2936      S  415ca60e            1     -  rotatelogs
 2938      S  415ca60e            1     -  rotatelogs
 3007      Z         0            1     -  sh
 3065      S  4151957e            1    S0  getty
 3458      S  41520eb8            1     1  vsh
 3607      Z         0            1     -  sh
 5622      Z         0            1     -  sh
16120      S  415293c6            1     -  httpd
16980      S  4151957e            1     1  login
17004      R   804b4d6            1     -  hwclock
17080      S  41523f92            1     -  dcos_sshd
17106      R  414c78d1            1     0  vsh
17211      S  414f2b0b            1     -  sleep
17226      S  414f2b0b            1     -  sleep
17227      S  4151957e            1     0  more
17228      S  41520494            1     0  vsh
17229      R  4151957e            1     -  ps
27264      S  415293c6            1     -  httpd
    -     NR         -            0     -  tacacs
    -     NR         -            0     -  bgp
    -     NR         -            0     -  dhcp_snoop
    -     NR         -            0     -  evb
    -     NR         -            0     -  installer
    -     NR         -            0     -  private-vlan
    -     NR         -            0     -  scheduler
    -     NR         -            0     -  vbuilder
    -     NR         -            0     -  vff
    -     NR         -            0     -  vtracker
```

Displaying CPU Utilization

Enter the **show processes cpu** command to display CPU utilization. (See Example 2-2.) The command output includes the following:

- Runtime(ms)—CPU time that the process has used, expressed in milliseconds.

- Invoked—Number of times that the process has been invoked.
- uSecs—Microseconds of CPU time as an average for each process invocation.
- 1Sec—CPU utilization as a percentage for the last one second.

Example 2-2 show processes cpu Command

```
n1000v# show processes cpu
PID    Runtime(ms)  Invoked   uSecs  1Sec    Process
-----  -----------  --------  -----  ------  -----------
    1        26994    364073     74    0.0%  init
    2            5       214     24    0.0%  kthreadd
    3         5861    174700     33    0.0%  migration/0
    4        17907   3927067      4    0.0%  ksoftirqd/0
    5          703     19374     36    0.0%  watchdog/0
    6         5315    155392     34    0.0%  migration/1
    7        16890   3767036      4    0.0%  ksoftirqd/1
    8           97     19374      5    0.0%  watchdog/1
    9       589280   1297793    454    0.0%  events/0
   10         4107    667550      6    0.0%  events/1
   11           93      1051     88    0.0%  khelper
   12          109      2029     53    0.0%  kblockd/0
   13         1812     44595     40    0.0%  kblockd/1
   14            0         2      0    0.0%  kacpid
   15            0         2      0    0.0%  kacpi_notify
   16            0        12     76    0.0%  kseriod
   17            0         2      9    0.0%  ata/0
   18            0         2      6    0.0%  ata/1
   19            0         2      1    0.0%  ata_aux
   20            0         2      0    0.0%  ksuspend_usbd
...
```

Displaying CPU and Memory Information

Enter the **show system resources** command to display system-related CPU and memory statistics. (See Example 2-3.) The output includes the following:

- The load is defined as the number of running processes. The average reflects the system load over the past 1, 5, and 15 minutes.
- Processes displays the number of processes in the system, and how many processes are actually running when the command is entered.
- CPU states shows the CPU usage percentage in the user mode, kernel mode, and idle time in the last one second.
- Memory usage provides the total memory, used memory, free memory, memory used for buffers, and memory used for cache in kilobytes. Buffers and cache are also included in the used memory statistics.

Example 2-3 show system resources Command

```
n1000v# show system resources
Load average:   1 minute: 0.50   5 minutes: 0.23   15 minutes: 0.13
Processes    :   299 total, 1 running
CPU states   :   1.0% user,   0.0% kernel,   99.0% idle
Memory usage:   4035420K total,   1280048K used,   2755372K free
Current memory status: OK
```

RADIUS

The RADIUS protocol is used for the exchange of attributes or credentials between a head-end RADIUS server and a client device. These attributes relate to three classes of services:

- Authentication
- Authorization
- Accounting

Authentication refers to the authentication of users for access to a specific device. You can use RADIUS to manage user accounts for access to a Cisco Nexus 1000V device. When you try to log in to a device, the Cisco Nexus 1000V validates you with information from a central RADIUS server.

Authorization refers to the scope of access that you have once you have been authenticated. Assigned roles for users can be stored in a RADIUS server with a list of actual devices that the user should have access to. Once the user has been authenticated, the switch can refer to the RADIUS server to determine the extent of access that the user will have within the switch network.

Accounting refers to the log information that is kept for each management session in a switch. This information can be used to generate reports for troubleshooting purposes and user accountability. Accounting can be implemented locally or remotely (using RADIUS).

This example shows how to display accounting log entries:

```
n1000v# show accounting log
Fri Aug 22 10:54:48 2014:type=stop:id=NSMGR:user=root:cmd=
Fri Aug 22 10:54:48 2014:type=start:id=NSMGR:user=root:cmd=
Fri Aug 22 10:54:48 2014:type=update:id=NSMGR:user=root:cmd=configure terminal ;
 port-profile "vmn_926e4512-f5e5-4639-8e2e-29344caf3dcc_654d375f-80b3-45d7-a8ea-
a370e30f4879" (SUCCESS)
Fri Aug 22 10:54:48 2014:type=update:id=NSMGR:user=root:cmd=configure terminal ;
 port-profile "vmn_926e4512-f5e5-4639-8e2e-29344caf3dcc_654d375f-80b3-45d7-a8ea-
a370e30f4879" ; switchport mode trunk (REDIRECT)
Fri Aug 22 10:54:48 2014:type=update:id=NSMGR:user=root:cmd=configure terminal ;
 port-profile "vmn_926e4512-f5e5-4639-8e2e-29344caf3dcc_654d375f-80b3-45d7-a8ea-
a370e30f4879" ; switchport mode trunk (SUCCESS)
Fri Aug 22 10:54:48 2014:type=stop:id=NSMGR:user=root:cmd=
Fri Aug 22 10:54:48 2014:type=start:id=NSMGR:user=root:cmd=
Fri Aug 22 10:54:48 2014:type=update:id=NSMGR:user=root:cmd=configure terminal ;
 port-profile "vmn_926e4512-f5e5-4639-8e2e-29344caf3dcc_654d375f-80b3-45d7-a8ea-
a370e30f4879" (SUCCESS)
Fri Aug 22 10:54:48 2014:type=update:id=NSMGR:user=root:cmd=configure terminal ;
 port-profile "vmn_926e4512-f5e5-4639-8e2e-29344caf3dcc_654d375f-80b3-45d7-a8ea-
a370e30f4879" ; switchport trunk allowed vlan all (REDIRECT)
Fri Aug 22 10:54:48 2014:type=update:id=NSMGR:user=root:cmd=configure terminal ;
 port-profile "vmn_926e4512-f5e5-4639-8e2e-29344caf3dcc_654d375f-80b3-45d7-a8ea-
a370e30f4879" ; switchport trunk allowed vlan all (SUCCESS)
```

Note The accounting log shows only the beginning and ending (start and stop) times for each session.

Syslog

The system message logging software saves messages in a log file or directs the messages to other devices. This feature provides the following capabilities:

- Logging information for monitoring and troubleshooting.
- Selecting the types of logging information to be captured.

- Selecting the destination of the captured logging information.

The syslog software allows you to store a chronological log of system messages locally or send to a central syslog server. Syslog messages can also be sent to the console for immediate use. These messages can vary in detail depending on the configuration that you choose.

Syslog messages are categorized into seven severity levels from *debug to critical* events. You can limit the severity levels that are reported for specific services within the switch.

Log messages are not saved across system reboots. However, a maximum of 100 log messages with a severity level of critical and below (levels 0, 1, and 2) can logged to a local file or server.

Logging Levels

The Cisco Nexus 1000V supports the following logging levels:

- 0—emergency
- 1—alert
- 2—critical
- 3—error
- 4—warning
- 5—notification
- 6—informational
- 7—debugging

By default, the switch logs normal but significant system messages to a log file and sends these messages to the system console. You can specify which system messages should be saved based on the type of facility and the severity level. Messages are time-stamped to enhance real-time debugging and management.

Enabling Logging for Telnet or SSH

System logging messages are sent to the console based on the default or configured logging facility and severity values.

You can disable logging to the console or enable logging to a given Telnet or SSH session as follows:

- To disable console logging, enter the **no logging console** command in global configuration mode.
- To enable logging for Telnet or SSH, enter the **terminal monitor** command in EXEC mode.

Note When logging to a console session that is disabled or enabled, that state is applied to all future console sessions. If you exit and log in again to a new session, the state is preserved. However, when logging to a Telnet or SSH session that is enabled or disabled, that state is applied only to that session. The state is not preserved after you exit the session.

The **no logging console** command that is shown in Example 2-4 disables console logging and is enabled by default.

Example 2-4 no logging console Command

```
n1000v(config)# no logging console
```

The **terminal monitor** command that is shown in Example 2-5 enables logging for Telnet or SSH and is disabled by default.

Example 2-5 terminal monitor Command

```
n1000v# terminal monitor
```

For more information about configuring syslogs, see the *Cisco Nexus 1000V for KVM System Management Configuration Guide, Release 5.x*.

High Availability

This chapter describes how to identify and resolve problems related to high availability.

Information About High Availability

The purpose of high availability (HA) is to limit the impact of failures—both hardware and software—within a system. The Cisco NX-OS operating system is designed for high availability at the network, system, and service levels.

The following Cisco NX-OS features minimize or prevent traffic disruption in the event of a failure:

- Redundancy—Redundancy at every aspect of the software architecture.
- Isolation of processes—Isolation between software components to prevent a failure within one process that is disrupting other processes.
- Restartability—Most system functions and services are isolated so that they can be restarted independently after a failure while other services continue to run. In addition, most system services can perform stateful restarts, which allow the service to resume operations transparently to other services.
- Supervisor stateful switchover— Active/standby dual supervisor configuration. The state and configuration remain constantly synchronized between two Virtual Switch Modules (VSMs) to provide a seamless and statefu1 switchover in the event of a VSM failure.

The Cisco Nexus 1000V system is made up of the following:

- Virtual Ethernet Modules (VEMs) that run within virtualization servers. The VEMs are represented as modules within the VSM.
- One or two VSMs that run within virtual machines (VMs).

Problems with High Availability

Symptom	Possible Causes	Solution
The active VSM does not see the standby VSM.	Roles are not configured properly. Check the role of the two VSMs by entering the **show system redundancy status** command.	1. Confirm that the roles are the primary and secondary role, respectively. 2. If needed, enter the **system redundancy role** command to correct the situation. 3. Save the configuration if roles are changed.
	Network connectivity problems. Check the L3 connectivity between the primary and secondary VSMs at the VSM host machines and upstream switches.	If network problems exist, do the following: 1. Shut down the VSM, which should be in standby mode. 2. Bring up the standby VSM after network connectivity is restored.
The active VSM does not complete synchronization with the standby VSM.	Version mismatch between VSMs. Check that the primary and secondary VSM are using the same image version by entering the **show version** of the command.	If the active and standby VSM software versions differ, reinstall the secondary VSM with the same version used in the primary.
	Fatal errors during gsync process. Check the gsyncctrl log by entering the **show system internal log sysmgr gsyncctrl** command and look for fatal errors.	Reload the standby VSM by entering the **reload module** *module-number* command, where *module-number* is the module number for the standby VSM.

Symptom	Possible Causes	Solution
The standby VSM reboots periodically.	The VSM has connectivity only through the management interface. When a VSM is able to communicate through the management interface, but not through the control interface, the active VSM detects the situation and resets the standby VSM to prevent the two VSMs from being in HA mode and out of sync. Check the output of the **show system internal redundancy info** command and verify if the *degraded_mode* flag is set to true.	Check the control port connectivity between the primary and secondary VSMs.
	VSMs have different versions. Enter the **debug system internal sysmgr all** command and look for the active_verctrl entry that indicates a version mismatch, as the following output shows: `2009 May  5 08:34:15.721920 sysmgr: active_verctrl: Stdby running diff version- force download the standby sup.`	Isolate the standby VSM and boot it. Enter the **show version** command to check the software version in both VSMs. Install the image matching the active VSM on the standby.
Both VSMs are in active mode.	Network connectivity problems. Check the L3 connectivity between the primary and secondary VSMs at the VSM host machines and upstream switches. When the VSM cannot communicate through any of these two interfaces, they will both try to become active.	If network problems exist, do the following: 1. Shut down the VSM, which should be in standby mode. 2. Bring up the standby VSM after network connectivity is restored.
	Different domain IDs in the two VSMs. Check the *domain* value by entering the **show system internal redundancy info** command.	If needed, update the domain ID and save it to the startup configuration. To upgrade the domain ID in a dual VSM system, do the following: 1. Isolate the VSM with the incorrect domain ID so that it cannot communicate with the other VSM. 2. Change the domain ID in the isolated VSM, save configuration, and power off the VSM. 3. Reconnect the isolated VSM and power it on.

System-Level High Availability

The Cisco Nexus 1000V supports redundant VSM VMs—a primary and a secondary—that run as an HA pair. Dual VSMs operate in an active/standby capacity in which only one of the VSMs is active at any given time, while the other acts as a standby backup. The state and configuration remain constantly synchronized between the two VSMs to provide a stateful switchover if the active VSM fails.

Single or Dual Supervisors

The Cisco Nexus 1000V system is made up of the following:

- VEMs that run within virtualization servers (these VEMs are represented as modules within the VSM)
- A remote management component, for example, the OpenStack dashboard.
- One or two VSMs that run within VMs.

Single VSM Operation	Dual VSM Operation
• Stateless—Service restarts from the startup configuration • Stateful—Service resumes from previous state.	• One active VSM and one standby VSM. • The active VSM runs all the system applications and controls the system. • On the standby VSM, the applications are started and initialized in standby mode. They are also synchronized and kept up to date with the active VSM in order to maintain the runtime context of “ready to run.” • On a switchover, the standby VSM takes over for the active VSM.

Network-Level High Availability

The Cisco Nexus 1000V HA at the network level includes port channels and the Link Aggregation Control Protocol (LACP). A port channel bundles physical links into a channel group to create a single logical link that provides the aggregate bandwidth of up to eight physical links. If a member port within a port channel fails, the traffic that was previously carried over the failed link switches to the remaining member ports within the port channel.

Additionally, the LACP allows you to configure up to 16 interfaces into a port channel. A maximum of eight interfaces can be active, and a maximum of eight interfaces can be placed in a standby state.

For additional information about port channels and the LACP, see the *Cisco Nexus 1000V for KVM Port Profile Configuration Guide, Release 5.x.*

High Availability Troubleshooting Commands

You can use the commands in this section to troubleshoot problems related to high availability.

To list process logs and cores, enter these commands:

- **show cores**

```
switch# show cores
Module  Instance  Process-name     PID       Date(Year-Month-Day Time)
------  --------  ---------------  --------  ------------------------
1       1        private-vlan      3207    Apr 28 13:29
```

- **show processes log [pid** *pid***]**

```
switch# show processes log
Process          PID     Normal-exit  Stack  Core   Log-create-time
---------------  ------  -----------  -----  -----  ---------------
private-vlan     3207              N      Y      N  Tue Apr 28 13:29:48 2009

switch# show processes log pid 3207
======================================================
Service: private-vlan
Description: Private VLAN

Started at Wed Apr 22 18:41:25 2009 (235489 us)
Stopped at Tue Apr 28 13:29:48 2009 (309243 us)
Uptime: 5 days 18 hours 48 minutes 23 seconds

Start type: SRV_OPTION_RESTART_STATELESS (23)
Death reason: SYSMGR_DEATH_REASON_FAILURE_SIGNAL (2) <-- Reason for the process abort
Last heartbeat 46.88 secs ago
System image name: switchh-dk9.5.2.1.SM15.0.1.bin
System image version: 5.2(1)SK1(1.1)

PID: 3207
Exit code: signal 6 (core dumped) <-- Indicates that a cores for the process was
generated.

CWD: /var/sysmgr/work
...
```

To check the redundancy status, enter this command:

- **show system redundancy status**

```
switch# show redundancy status
Redundancy role
---------------
      administrative:   primary
         operational:   primary

Redundancy mode
---------------
      administrative:   HA
         operational:   None

This supervisor (sup-1)
-----------------------
    Redundancy state:   Active
    Supervisor state:   Active
      Internal state:   Active with no standby

Other supervisor (sup-2)
------------------------
    Redundancy state:   N/A

    Supervisor state:   N/A
      Internal state:   N/A

System start time:          Thu Sep  4 16:48:55 2014

System uptime:              7 days, 11 hours, 39 minutes, 24 seconds
Kernel uptime:              7 days, 11 hours, 39 minutes, 11 seconds
Active supervisor uptime:   7 days, 11 hours, 38 minutes, 45 seconds
```

To check the system internal redundancy status, enter this command:

- **show system internal redundancy info**

```
switch# show system internal redundancy info
My CP:
  slot: 0
  domain: 36
  role:   primary
  status: RDN_ST_AC
  state:  RDN_DRV_ST_AC_NP
  intr:   enabled
  power_off_reqs: 0
  reset_reqs:     1
  inter_vsm_max_heartbeat_loss:  15
  product_type:  2
Other CP:
  slot: 1
  status: RDN_ST_NP
  active: true
  ver_rcvd: false
  degraded_mode: true
  prod_type rcvd: false
  peer mac rcvd: false
Redun Device 0:
  name: ha0
  pdev: c9949800
  alarm: false
  mac: ff:ff:ff:ff:ff:ff
  tx_set_ver_req_pkts:   646867
  tx_set_ver_rsp_pkts:   0
  tx_peer_mac_req_pkts: 0
  tx_peer_mac_rsp_pkts: 0
  tx_heartbeat_req_pkts: 0
  tx_heartbeat_rsp_pkts: 0
  rx_set_ver_req_pkts:   0
  rx_set_ver_rsp_pkts:   0
  rx_peer_mac_req_pkts:  0
  rx_peer_mac_rsp_pkts:  0
  rx_heartbeat_req_pkts: 0
  rx_heartbeat_rsp_pkts: 0
  rx_drops_wrong_domain: 0
  rx_drops_wrong_slot:   0
  rx_drops_short_pkt:    0
  rx_drops_queue_full:   0
  rx_drops_inactive_cp:  0
  rx_drops_bad_src:      0
  rx_drops_not_ready:    0
  rx_drops_wrong_ver:    0
  rx_unknown_pkts:       0
  tx_rdn_mgr_params_msg_pkts:  0
  tx_rdn_mgr_params_ack_pkts:  0
  rx_rdn_mgr_params_msg_pkts:  0
  rx_rdn_mgr_params_ack_pkts:  0
Redun Device 1:
  name: ha1
  pdev: c994c800
  alarm: false
  mac: ff:ff:ff:ff:ff:ff
  tx_set_ver_req_pkts:   646867
  tx_set_ver_rsp_pkts:   0
  tx_peer_mac_req_pkts: 0
  tx_peer_mac_rsp_pkts: 0
  tx_heartbeat_req_pkts: 0
  tx_heartbeat_rsp_pkts: 0
  rx_set_ver_req_pkts:   0
  rx_set_ver_rsp_pkts:   0
  rx_peer_mac_req_pkts:  0
  rx_peer_mac_rsp_pkts:  0
  rx_heartbeat_req_pkts: 0
  rx_heartbeat_rsp_pkts: 0
  rx_drops_wrong_domain: 0
  rx_drops_wrong_slot:   0
  rx_drops_short_pkt:    0
  rx_drops_queue_full:   0
  rx_drops_inactive_cp:  0
  rx_drops_bad_src:      0
  rx_drops_not_ready:    0
  rx_drops_wrong_ver:    0
  rx_unknown_pkts:       0
  tx_rdn_mgr_params_msg_pkts:  0
  tx_rdn_mgr_params_ack_pkts:  0
  rx_rdn_mgr_params_msg_pkts:  0
  rx_rdn_mgr_params_ack_pkts:  0
```

To check the system internal sysmgr state, enter this command:

- **show system internal sysmgr state**

```
switch# show system internal sysmgr state
The master System Manager has PID 1323 and UUID 0x1.
Last time System Manager was gracefully shutdown.
The state is SRV_STATE_MASTER_ACTIVE_ALONE entered at time Thu Sep  4 16:49:07 2
014.

The '-b' option (disable heartbeat) is currently disabled.

The '-n' (don't use rlimit) option is currently disabled.

Hap-reset is currently enabled.

Process restart capability is currently disabled.

Watchdog checking is currently enabled.

Watchdog kgdb setting is currently enabled.


        Debugging info:

The trace mask is 0x00000000, the syslog priority enabled is 3.
The '-d' option is currently disabled.
The statistics generation is currently enabled.


        HA info:

slotid = 1    supid = 0
cardstate = SYSMGR_CARDSTATE_ACTIVE .
cardstate = SYSMGR_CARDSTATE_ACTIVE (hot switchover is configured enabled).
Configured to use the real platform manager.
Configured to use the real redundancy driver.
Redundancy register: this_sup = RDN_ST_AC, other_sup = RDN_ST_NP.
EOBC device name: eth0.
Remote addresses:  MTS - [not available]      IP - [not available]
MSYNC not done.
Remote MSYNC not done.
Module online notification received.
```

```
Local super-state is: SYSMGR_SUPERSTATE_STABLE
Standby super-state is: SYSMGR_SUPERSTATE_STABLE
Swover Reason : SYSMGR_UNKNOWN_SWOVER
Total number of Switchovers: 0
Swover threshold settings: 20 switchovers within 1200 seconds
Switchovers within threshold interval: 0
Last switchover time: 0 seconds after system start time
Cumulative time between last 0 switchovers: 0
Start done received for 3 plugins, Total number of plugins = 3


        Statistics:

Message count:           0
Total latency:           0             Max latency:             0
Total exec:              0             Max exec:                0
```

To reload a module, enter this command:

- **reload module**

  ```
  switch# reload module 2
  ```

  This command reloads the secondary VSM.

Note Entering the **reload** command without specifying a module reloads the whole system.

To attach to the standby VSM console, enter this command:

- **attach module**

  The standby VSM console is not accessible externally but can be accessed from the active VSM through the **attach module** *module-number* command.

  ```
  switch# attach module 2
  ```

  This command attaches to the console of the secondary VSM.

CHAPTER 4

VSM and VEM Modules

This chapter describes how to identify and resolve problems that relate to modules.

Information About Modules

The Cisco Nexus 1000V implementation has two parts:

- Virtual Supervisor Module (VSM)—This is the control software of the Cisco Nexus 1000V distributed virtual switch. It runs on a VM and is based on Cisco NX-OS software.
- Virtual Ethernet Module (VEM)—This is the part of Cisco Nexus 1000V that actually switches data traffic. It runs on a KVM (Kernel-based virtual machine) server. Several VEMs are controlled by one VSM.

Troubleshooting a Module That Does Not Come Up on the VSM

This section describes the process that you can use when a module does not come up on the VSM.

Troubleshooting Guidelines

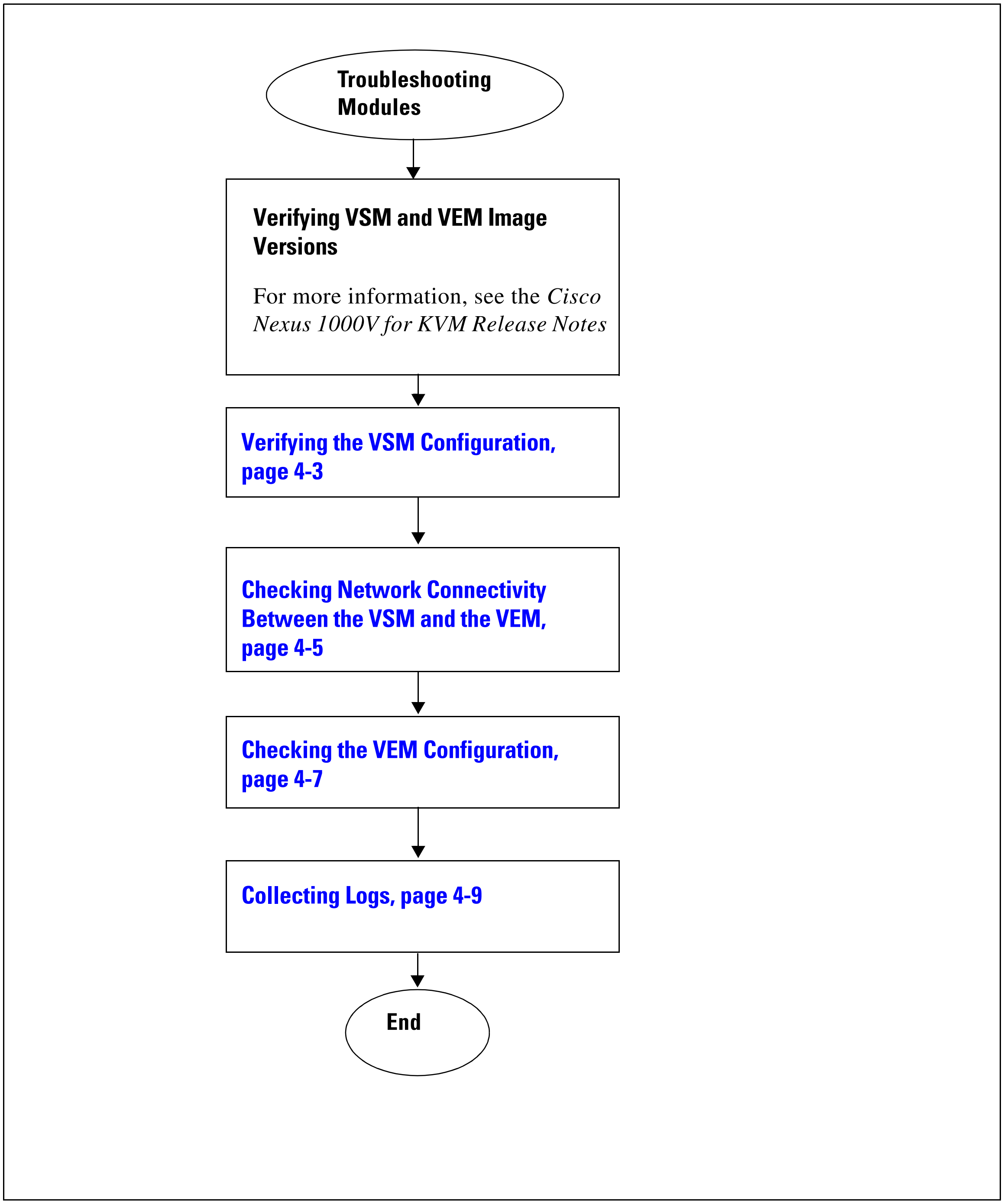
Follow these guidelines when troubleshooting a module that is controlled by the VSM:

- You must have a VSM VM and a VEM up and running.
- Make sure that you are running compatible versions of the OpenStack server and VSM.
- Make sure that the VEM has reachability to the VSM.

  For more information, see the *Cisco Nexus 1000V for KVM Release Notes*.

Flowchart for Troubleshooting Modules

Use the following flowchart to troubleshoot modules.


Troubleshooting Modules
Verifying VSM and VEM Image Versions
For more information, see the *Cisco Nexus 1000V for KVM Release Notes*
Verifying the VSM Configuration, page 4-3
Checking Network Connectivity Between the VSM and the VEM, page 4-5
Checking the VEM Configuration, page 4-7
Collecting Logs, page 4-9
End

Verifying the VSM Configuration

You can verify the domain configuration.

Step 1 Log in to the CLI in EXEC mode.

Step 2 On the VSM, verify the domain configuration by entering this command:

show svs domain

```
n1000v# show svs domain
SVS domain config:
  Domain id:    36
  Control vlan:  NA
  Packet vlan:   NA
  Control mode: L3
  Switch guid: 6bd22a84-b262-4327-8bd0-696109748c6a
  L3 control interface: mgmt0
  Status: Config not pushed to Management Server.
  Control type multicast: No
Note: Control VLAN and Packet VLAN are not used in L3 mode
```

Problems with the VSM

The following are symptoms, possible causes, and solutions for problems with the VSM.

Symptom	Possible Causes	Solution
After a VSM is rebooted, the system stops functioning in one of the following states and does not recover on its own. Attempts to debug fail.		
After boot, the VSM has a loader prompt.	The VSM kickstart image has been corrupted.	1. Disable the primary and secondary VSM resources in the pacemaker cluster. 2. Log in to the nodes with active VSMs and shut down the VMs. 3. Log in to the controller nodes and run the **qemu image** command. 4. Enable the primary and secondary VSMs in the pacemaker cluster. 5. Verify the active and standby VSMs using the show module command. 6. Compare the running configuration with the configuration defined in the backup file. Use show running-config command to view the running configuration. If there are any discrepancies between the running configuration and backup configuration, run the missing configuration commands on the VSM. For detailed information, see *Cisco Nexus 1000V for KVM Installation Guide for Red Hat Enterprise Linux OpenStack Platform 7*.
After boot, the VSM has a boot prompt.	The VSM system image has been corrupted.	1. Disable the primary and secondary VSM resources in the pacemaker cluster. 2. Log in to the nodes with active VSMs and shutdown the VMs. 3. Log in to the controller nodes and run the **qemu image** command. 4. Enable the primary and secondary VSMs in the pacemaker cluster. 5. Verify the active and standby VSMs using the show module command. 6. Compare the running configuration with the configuration defined in the backup file. Use show running-config command to view the running configuration. If there are any discrepancies between the running configuration and backup configuration, run the missing configuration commands on the VSM. For detailed information, see *Cisco Nexus 1000V for KVM Installation Guide for Red Hat Enterprise Linux OpenStack Platform 7*.

Symptom	Possible Causes	Solution
After boot, the VSM has been reconfigured.	The startup configuration has been deleted.	Do one of the following: • If you have a saved backup copy of your configuration file, restore the configuration on the VSM by entering the **copy** *source-filessystem:filename* **running-config** command. If a backup copy of the running configuration is not available, contact TAC for advanced recovery procedures.

After boot, the VSM stopped at "Loader Loading."	The boot menu file has been corrupted.	1. Disable both the primary and secondary VSM resources in pacemaker.
		2. Log in to nodes with active VSMs and shutdown the VMs.
		3. Log in to all the three controllers and run the following commands:
		`[root@overcloud-controller-0 heat-admin]# qemu-img create /var/spool/cisco/vsm/primary_disk 4G`
		`[root@overcloud-controller-0 heat-admin]# qemu-img create /var/spool/cisco/vsm/secondary_disk 4G`
		`[root@overcloud-controller-1 heat-admin]# qemu-img create /var/spool/cisco/vsm/primary_disk 4G`
		`[root@overcloud-controller-1 heat-admin]# qemu-img create /var/spool/cisco/vsm/secondary_disk 4G`
		`[root@overcloud-controller-2 heat-admin]# qemu-img create /var/spool/cisco/vsm/primary_disk 4G`
		`[root@overcloud-controller-2 heat-admin]# qemu-img create /var/spool/cisco/vsm/secondary_disk 4G`
		4. Enable both the primary and secondary VSMs in pacemaker.
		5. Log in to the primary VSM and verify both the active and standby VSMs using the **show module** command.
		6. Compare the running configuration with the configuration defined in the backup file. Use **show running-config** command to view the running configuration. If there are any discrepancies between the running configuration and backup configuration, run the missing configuration commands on the VSM.

After boot, the secondary VSM reboots continuously.	Check the L3 connectivity between the two VSMs.	Check the control connectivity between the active and standby VSM.
	Active and standby VSMs are failing to synchronize.	From the active VSM, check gsyncstats to identify which application caused the failure by entering the **show logging** command.
The management IP address is not reachable.	Varies.	Use the **virsh console** (*$vsm-vm-name*) command on the VSM host node to connect directly to the console.

Checking Network Connectivity Between the VSM and the VEM

You can verify Layer 2 network connectivity between the VSM and the VEM.

Step 1 On the VSM, find its MAC address by entering this command:

show svs neighbors

The VSM MAC address displays as the AIPC Interface MAC.

The user VEM Agent MAC address of the host displays as the Src MAC.

```
n1000v# show svs neighbors

Active Domain ID: 36

AIPC Interface MAC: 5254-0040-9ad6
Inband Interface MAC: 5254-0002-3a0f

Src MAC           Type   Domain-id    Node-id     Last learnt (Sec. ago)
------------------------------------------------------------------------

0002-3d40-2403     VEM        36         0402          0.07
0002-3d40-2404     VEM        36         0502          0.07
0002-3d40-2405     VEM        36         0602          0.07
```

Step 2 Do one of the following:

- If the output of the **show svs neighbors** command in Step 1 does not display the VEMs, there might be a problem with the VSM network connectivity. Proceed to the next step.
- If only some VEMs are missing, the problem might be on the VEM. See Checking the VEM Configuration, page 4-7.

Step 3 On the upstream switch, display the MAC address table to verify the network configuration by entering this command:

show mac address-table interface *int_id* **vlan** *vlan_id*

Note The MAC address table should be checked on the VLAN where the VSM is connected.

```
switch# show mac address-table interface Gi3/2 vlan 3002
Legend: * - primary entry
        age - seconds since last seen
        n/a - not available

  vlan   mac address     type    learn     age              ports
------+----------------+--------+-----+----------+--------------------------
Active Supervisor:
* 3002  00:02:3d:40:0b:0c   dynamic  Yes          0   Gi3/2
```

Step 4 If the output from Step 3 does not display the MAC address of the VSM, there might be a problem with the VSM's network connectivity.

Verifying the VEM Installation

Step 1 Verify the VEM installation by entering the **show svs upgrade status** command

```
n1000v# show svs upgrade status
Upgrade State: Active
Upgrade mgmt0 ipv4 addr:
Upgrade mgmt0 ipv6 addr:
Upgrade control0 ipv4 addr:
```

Step 2 Check that the upgrade state is active and does not report any errors.

Checking the VEM Configuration

You can verify the VEM configuration.

Step 1 Verify the domain ID by entering the **vemcmd show card** command:

```
n1000v# attach vem 3
n1000v(vem-attach)# vemcmd show card
Card UUID type  2: 3EC25838-8116-11E4-0000-00000000011F
Card name: mac0025b50d005f.example.com
Switch name: VSM1-P
Switch alias: NA
Switch uuid: c3ca0345-770c-4c84-9731-dbd49d62c095
Card domain: 501
Card slot: 3
VEM Tunnel Mode: L3 Mode
L3 Ctrl Index: 0
VEM Control (AIPC) MAC: 00:02:3d:11:f5:02
VEM Packet (Inband) MAC: 00:02:3d:21:f5:02
VEM Control Agent (DPA) MAC: 00:02:3d:41:f5:02
VEM SPAN MAC: 00:02:3d:31:f5:02
Primary VSM MAC : 00:50:66:ee:04:00
Primary VSM PKT MAC : 00:00:00:00:00:00
Primary VSM MGMT MAC : 00:00:00:00:00:00
Standby VSM CTRL MAC : 00:00:00:00:00:00
Management IPv4 address: 11.11.0.22
Management IPv6 address: 0000:0000:0000:0000:0000:0000:0000:0000
Primary L3 Control IPv4 address: 0.0.0.0
Secondary VSM MAC : 00:00:00:00:00:00
Upgrade : Default
Max physical ports: 32
Max virtual ports: 990
Card control VLAN: 0
Card packet VLAN: 0
Control type multicast: No
Card Headless Mode : No
DPA Status : Up
       Processors: 6
  Processor Cores: 6
Processor Sockets: 2
  Kernel Memory:   0
Port link-up delay: 5s
Global UUFB: DISABLED
Layer 3 Forwarding: DISABLED
Heartbeat Set: True
Card Type: vem
PC LB Algo: source-mac
Datapath portset event in progress : no
Licensed: Yes
Global BPDU Guard: Disabled
DP Initialized: Yes
Tag Native VLAN: No
L3Sec Mode: TRUE
Layer 3 Forwarding mode: Gateway-Mode
Layer 3 Forwarding Mac : 00:02:3d:b0:00:00
n1000v(vem-attach)#
```

Step 2 Verify that the ports of the host added to the logical switch are listed and that the ports are correctly configured as access or trunk on the host by entering the **vemcmd show port** command:

```
VSM# module vem 3 execute vemcmd show port-old
LTL    IfIndex   Vlan/    Bndl  SG_ID Pinned_SGID  Type  Admin State      CBL Mode   Name
                  SegId
    6          0      1 T     0     32          32  VIRT     UP    UP    1  Trunk vns
   11          0   3968       0     32          32  VIRT     UP  DOWN    1 Access _l23
   12          0      1       0     32           1  VIRT     UP  DOWN    1 Access _l24
   13          0      1       0     32          32  VIRT     UP  DOWN    0 Access _l25
   15          0   3971       0     32          32  VIRT     UP  DOWN    1 Access _l27
   16          0      1 T     0     32          32  VIRT     UP  DOWN    1  Trunk arp
   17          0      1       0     32          32  VIRT     UP    UP    0 Access _l2vxen
   18   2500c000      1 T  1040      0          32  PHYS     UP    UP    1  Trunk eth1
   19   2500c040      1 T  1040      1          32  PHYS     UP    UP    1  Trunk eth0
   50   1c000050     40       0     32           0  VIRT     UP    UP    1 Access
cn1-vtep1-ovs
 1040   16000002      1 T     0     32          32  CHAN     UP    UP    1  Trunk
```

The last line of the output indicates that vmnic1 should be in trunk mode with a color blocking logic (CBL) value of 1. The CBL value of the native VLAN does not have to be 1. It can be 0 if it is not allowed or 1 if it is VLAN 1 and not allowed. If the CBL value is 0 it is not a problem unless the native VLAN is the control VLAN. The Admin state and Port state should be UP.

Step 3 Check if the VSM is reachable from the OpenStack host by entering these commands:

```
route
arp -a
```

Problems with the VEM

The following are symptoms, possible causes, and solutions for problems with the VEM.

Symptom	Possible Causes	Solution
A VEM that you created has failed.	Check whether the OpenStack services are installed. For the compute node, the nova-compute service has to be installed before installing the VEM.	1. Ensure that the VSM/VEM are installed into the Overcloud image. 2. Run **glance image-list** command on the Undercloud. 3. Verify that the size of overcloud-full image matches the size of the modified Overcloud image (with VSM/VEM installed). If the sizes do not match, remove the five images using **glance image-delete** (*$image-uuid*) command and re-install the images using **openstack overcloud image upload** *--image-path /home/stack/images/* command. 4. Run the **openstack baremetal configure boot** command to update the image configuration for all node types. Verify that the redeployed Overcloud is using modified image.
	VEM is installed and VSM is reachable from VEM but still the module is not attached. This condition occurs when the VSM and VEM are in different subnet.	1. Ensure that the N1000vVEMHostMgmtIntf configuration is on the same network as N1000vVSMHostMgmtIntf. 1. Synchronize the configuration across all the VEM nodes using the **deploy** command. If the VEM configuration is different between Control and Compute nodes, use the per-node configuration parameter to configure the control nodes because there are three Controller nodes and the Compute nodes may be added later.

Collecting Logs

After you verify the network connectivity between the VEM and the VSM, you can collect log files to help identify the problem.

Step 1 On the VEM, verify its universally unique identifier (UUID) by entering the **vemcmd show card info** command:

```
n1000v# vemcmd show card info
Card UUID type  2: 6AC6E608-C51D-E211-0010-20304050008D
Card name: compute-1
Switch name: vsm-p
Switch alias: NA
Switch uuid: 6bd22a84-b262-4327-8bd0-696109748c6a
Card domain: 36
Card slot: 4
VEM Tunnel Mode: L3 Mode
L3 Ctrl Index: 0
VEM Control (AIPC) MAC: 00:02:3d:10:24:03
VEM Packet (Inband) MAC: 00:02:3d:20:24:03
VEM Control Agent (DPA) MAC: 00:02:3d:40:24:03
VEM SPAN MAC: 00:02:3d:30:24:03
Primary VSM MAC : 52:54:00:40:9a:d6
Primary VSM PKT MAC : 00:00:00:00:00:00
Primary VSM MGMT MAC : 00:00:00:00:00:00
Standby VSM CTRL MAC : 00:00:00:00:00:00
Management IPv4 address: 172.27.0.215
Management IPv6 address: 0000:0000:0000:0000:0000:0000:0000:0000
Primary L3 Control IPv4 address: 0.0.0.0
Secondary VSM MAC : 00:00:00:00:00:00
Upgrade : Default
Max physical ports: 32
Max virtual ports: 990
Card control VLAN: 0
Card packet VLAN: 1
Control type multicast: No
Card Headless Mode : No
DPA Status : Up
       Processors: 6
  Processor Cores: 6
Processor Sockets: 2
  Kernel Memory:   0
Port link-up delay: 5s
Global UUFB: DISABLED
Heartbeat Set: True
Card Type: vem
PC LB Algo: source-mac
Datapath portset event in progress : no
Licensed: Yes
Global BPDU Guard: Disabled
DP Initialized: Yes
Tag Native VLAN: No
L3Sec Mode: TRUE
```

Step 2 On the VSM, verify the module number to which the corresponding UUID entry is mapped by entering the **show module vem mapping** command:

```
n1000v# show module vem mapping
Mod     Status         UUID                                    License Status
---     -----------    ------------------------------------    --------------
3       absent         6AC6E608-C51D-E211-0010-20304050005E    unlicensed
4       powered-up     6AC6E608-C51D-E211-0010-20304050008D    licensed
5       powered-up     6AC6E608-C51D-E211-0010-20304050001E    licensed
6       powered-up     6AC6E608-C51D-E211-0010-2030405000AD    licensed
n1000v#
```

Step 3 Using the module number from Step 2, collect the output of these commands:

- **show platform internal event-history module 13**
- **show module internal event-history module 13**
- **show system internal im event-history module 13**
- **show system internal vmm event-history module 13**
- **show system internal ethpm event-history module 13**

If you need to contact Cisco TAC for assistance in resolving an issue, you must have the output of the commands listed in Step 3.

VSM and VEM Troubleshooting Commands

You can use the commands in this section to troubleshoot problems related to the VSM and the VEM. Enter these commands in a PowerShell window > Linux command prompt.

Note VSM commands should be issued from within an SSH session to the VSM. VEM commands should be issued on the command prompt of the control or compute node that you are troubleshooting.

VSM Command	Description
show svs neighbors	Displays all SVS neighbors. See Example 4-1 on page 4-12.
show svs domain	Displays the domain configuration. See Example 4-2 on page 4-13.
show port-profile name *name*	Displays the configuration for a named port profile. See Example 4-3 on page 4-13.
show running-config vlan *vlanID*	Displays the VLAN information in the running configuration. See Example 4-4 on page 4-14.
show mac address-table interface	Displays the MAC address table on an upstream switch to verify the network configuration. See Example 4-5 on page 4-14.
module vem *module_number* **execute vemcmd show l2** [*control_vlan_id* | *packet_vlan_id]*	Displays the VLAN configuration on the VEM to verify that the VSM MAC address appears in the control and packet VLANs. See Example 4-6 on page 4-14.
vemlog	Displays and controls the VEM kernel logs.
vemlog show last *number-of-entries*	Displays the circular buffer. See Example 4-21 on page 4-22.
vemlog show info	Displays information about entries in the log. See Example 4-22 on page 4-22.
vem-support.ps1	Navigate to the support directory under \Nexus1000V and run the vem-support.ps1 script. See Example 4-24 on page 4-28.
show module vem mapping	Displays information about the VEM that a VSM maps to, including the VEM module number, status, UUID, and license status. See Example 4-14 on page 4-17.
show platform internal event-history module *module-number*	Displays platform FSM event information.
show module internal event-history module *module-number*	Displays the event log for a module. See Example 4-15 on page 4-17.

VSM Command	Description
show system internal im event-history module *module-number*	Displays the module IM event logs for the system. See Example 4-16 on page 4-18.
show system internal vmm event-history module *module-number*	Displays the module VMM event logs for the system. See Example 4-17 on page 4-19.
show system internal ethpm event-history module *module-number*	Displays the module Ethernet event logs for the system. See Example 4-18 on page 4-20.
show system internal ethpm event-history interface *type slot*	Displays the Ethernet interface logs for the system. See Example 4-19 on page 4-21.

VEM Command	Description
vemcmd	Displays configuration and status information.
vemcmd show card	Displays information about the cards on the VEM to verify that the domain ID, control VLANs, and packet VLANs are configured correctly on the host. See Example 4-6 on page 4-14.
vemcmd show attach	Displays information about the platform port attach. See Example 4-9 on page 4-15.
vemcmd show vem internal info	Displays information about the VEM queue status. See Example 4-10 on page 4-15.
vemcmd show port [*port-LTL-number*]	Displays information about the ports on the VEM to verify that the ports of the host added to the DVS are listed and that the ports are correctly configured as access or trunk on the host. See Example 4-11 on page 4-16
vemcmd show bd [*control_vlan_id* | *packet_vlan_id]*	Displays the list of ports that belong to the VLAN. The bd number is not the same as a VLAN number. You can display a listing of bd numbers by entering the **vemcmd show bd** command. See Example 4-12 on page 4-16.
vemcmd show trunk	Displays configured information about the VEM to verify that the DV port groups are successfully pushed from the KVM server to the host and that the correct physical trunk port VM NIC is used. See Example 4-13 on page 4-16.
vemcmd show version	Displays the version information. See Example 4-20 on page 4-21.
vemcmd help	Displays the type of information you can display. See Example 4-23 on page 4-22.

Example 4-1 show svs neighbors command

```
switch# show svs neighbors
```

```
Active Domain ID: 36

AIPC Interface MAC: 5254-0040-9ad6
Inband Interface MAC: 5254-0002-3a0f

Src MAC           Type   Domain-id    Node-id     Last learnt (Sec. ago)
------------------------------------------------------------------------

0002-3d40-2403     VEM        36         0402         0.49
0002-3d40-2404     VEM        36         0502         0.49
0002-3d40-2405     VEM        36         0602         0.49

switch#
```

Example 4-2 show svs domain command

```
n1000v# show svs domain

SVS domain config:
  Domain id:    36
  Control vlan:  NA
  Packet vlan:   NA
  Control mode: L3
  Switch guid: 6bd22a84-b262-4327-8bd0-696109748c6a
  L3 control interface: mgmt0
  Status: Config not pushed to Management Server.
  Control type multicast: No

Note: Control VLAN and Packet VLAN are not used in L3 mode
```

Example 4-3 show port-profile command

```
switch# show port-profile name SystemUplink
port-profile SystemUplink
type: Ethernet
 description: NSM created profile. Do not delete.
 status: enabled
 max-ports: 512
 min-ports: 1
 inherit: PortChannelProfile
 config attributes:
  switchport mode trunk
  switchport trunk allowed vlan 173
 evaluated config attributes:
  switchport mode trunk
  switchport trunk allowed vlan 173
  channel-group auto
  no shutdown
 assigned interfaces:
 port-group:
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vn-service: no
 port-profile role: none
 port-binding: static
```

Example 4-4 show running-config vlan command

```
switch# show running-config vlan 260-261
!Time: Fri Sep 12 05:12:34 2014

version 5.2(1)SK3(2.1)
#
```

Example 4-5 show mac address-table interface command

```
switch# show mac address-table interface Gi3/1 vlan 3002
Legend: * - primary entry
        age - seconds since last seen
        n/a - not available

  vlan   mac address     type    learn     age              ports
------+----------------+--------+-----+----------+-------------------------
Active Supervisor:
* 3002  0050.56be.7ca7   dynamic  Yes          0   Gi3/1
```

Example 4-6 vemcmd show card info command

```
switch# vemcmd show card info
Card UUID type  2: 6AC6E608-C51D-E211-0010-20304050008D
Card name: compute-1
Switch name: vsm-p
Switch alias: NA
Switch uuid: 6bd22a84-b262-4327-8bd0-696109748c6a
Card domain: 36
Card slot: 4
VEM Tunnel Mode: L3 Mode
L3 Ctrl Index: 0
VEM Control (AIPC) MAC: 00:02:3d:10:24:03
VEM Packet (Inband) MAC: 00:02:3d:20:24:03
VEM Control Agent (DPA) MAC: 00:02:3d:40:24:03
VEM SPAN MAC: 00:02:3d:30:24:03
Primary VSM MAC : 52:54:00:40:9a:d6
Primary VSM PKT MAC : 00:00:00:00:00:00
Primary VSM MGMT MAC : 00:00:00:00:00:00
Standby VSM CTRL MAC : 00:00:00:00:00:00
Management IPv4 address: 172.27.0.215
Management IPv6 address: 0000:0000:0000:0000:0000:0000:0000:0000
Primary L3 Control IPv4 address: 0.0.0.0
Secondary VSM MAC : 00:00:00:00:00:00
Upgrade : Default
Max physical ports: 32
Max virtual ports: 990
Card control VLAN: 0
Card packet VLAN: 1
Control type multicast: No
Card Headless Mode : No
DPA Status : Up
       Processors: 6
  Processor Cores: 6
Processor Sockets: 2
  Kernel Memory:   0
Port link-up delay: 5s
Global UUFB: DISABLED
Heartbeat Set: True
Card Type: vem
PC LB Algo: source-mac
Datapath portset event in progress : no
```

```
Licensed: Yes
Global BPDU Guard: Disabled
DP Initialized: Yes
Tag Native VLAN: No
L3Sec Mode: TRUE
```

Example 4-7 vemcmd show vmq allocation command

```
~ # vemcmd show vmq allocation
LTL   VSM Port  Phy LTL  Queue id  Team queue id
   49     Veth13       17        1       49
                 18   2               49
   50     Veth14       17        2       50
                 18   3               50
   51     Veth16       19        1       51
                 20   1               51
```

Example 4-8 vemcmd show vmq resources command

```
~ # vemcmd show vmq resources
LTL   VSM Port  Max queues  Free queues
   17     Eth3/1          16           10
   18     Eth3/2          16           10
   19     Eth3/3           8            7
```

Example 4-9 vemcmd show attach command

```
~ # vemcmd show attach
----------------------------------------------
LTL:              17
----------------------------------------------
Port ID:          1
NIC Index:        1
Port UUID:        BC9C4957-88B0-4292-879A-A4109A5A345B
NIC Instance ID:  {239C8D0D-43AD-4DB7-94E1-1D90D265D21F}
MAC address:      d0:d0:fd:09:31:f8
Port profile:     uplink-trunk
VM/NIC name:      Intel(R) 82576 Gigabit Dual Port Network Connection
VM UUID:
MTU:              1514
Link state:       UP
Duplex:           Full
Tx speed:         1000000000
Rx speed:         1000000000
Autoneg:          Enabled
Link Params pending: No
Speed Capability 0x13
Duplex Capability 0x7
```

Example 4-10 vemcmd show vem internal info command

```
~ # vemcmd show vem internal info
----------------------------------------------
VEM Internal counters
----------------------------------------------
# Tx pkts pending:             0
# Timer events queued:         0
# Internal pkts queued:        0
# DPA notifications queued:    0
```

Example 4-11 vemcmd show port command

```
vsm-p(vem-attach)# vemcmd show vem internal info command
vsm-p(vem-attach)# vemcmd show vem internal info
vsm-p(vem-attach)# vemcmd show port
  LTL   VSM Port  Admin Link  State  PC-LTL  SGID          Vem Port  Type      ORG
svcpath Owner
   18     Eth4/1     UP   UP    FWD    1040     0              eth1              0
0
   19     Eth4/2     UP   UP    FWD    1040     1              eth0              0
0
   50      Veth6     UP   UP    FWD       0     0     cn1-vtep1-ovs  VXLAN       0
0
 1040        Po3     UP   UP    FWD       0                                      0
0

* F/B: Port is BLOCKED on some of the vlans.
       One or more vlans are either not created or
       not in the list of allowed vlans for this port.
 Please run "vemcmd show port vlans" to see the details.
```

Example 4-12 vemcmd show bd command

```
VSM# module vem 3 execute vemcmd show bd 8
BD 1, vdc 1, vlan 1, swbd 1, 4 ports, ""

Portlist:
     12  _l24
     18  eth1
     19  eth0
   1040

BD 2, vdc 1, vlan 3972, swbd 3972, 0 ports, ""

Portlist:
BD 3, vdc 1, vlan 3970, swbd 3970, 0 ports, ""

Portlist:
BD 4, vdc 1, vlan 3968, swbd 3968, 3 ports, ""

Portlist:
      1  inband
      5  inband port security
     11  _l23

BD 5, vdc 1, vlan 3971, swbd 3971, 1 ports, ""

Portlist:
     15  _l27

BD 6, vdc 1, vlan 40, swbd 40, 4 ports, ""

Portlist:
     18  eth1
     19  eth0
     50  cn1-vtep1-ovs
   1040
```

Example 4-13 vemcmd show trunk command

```
~ # vemcmd show trunk
```

```
Trunk port 6 native_vlan 1 CBL 1
vlan(1) cbl 1, vlan(3972) cbl 1, vlan(3970) cbl 1, vlan(3968) cbl 1, vlan(3971) cbl 1,
vlan(40) cbl 1,
Trunk port 16 native_vlan 1 CBL 1
vlan(1) cbl 1, vlan(3972) cbl 1, vlan(3970) cbl 1, vlan(3968) cbl 1, vlan(3971) cbl 1,
vlan(40) cbl 1,
Trunk port 18 native_vlan 1 CBL 1
vlan(1) cbl 1, vlan(40) cbl 1,
Trunk port 19 native_vlan 1 CBL 1
vlan(1) cbl 1, vlan(40) cbl 1,
Trunk port 1040 native_vlan 1 CBL 1
vlan(1) cbl 1, vlan(40) cbl 1
```

Example 4-14 show module vem mapping command

```
switch# show module vem mapping
Mod     Status          UUID                                    License Status
---     -----------     ------------------------------------    --------------
3       absent          6AC6E608-C51D-E211-0010-20304050005E    unlicensed
4       powered-up      6AC6E608-C51D-E211-0010-20304050008D    licensed
5       powered-up      6AC6E608-C51D-E211-0010-20304050001E    licensed
6       powered-up      6AC6E608-C51D-E211-0010-2030405000AD    licensed#
```

Example 4-15 show module internal event-history module command

```
switch# show module internal event-history module 1
>>>>FSM: <ID(257): Slot 1, node 0x0101> has 16 logged transitions<<<<<

1) FSM:<ID(257): Slot 1, node 0x0101> Transition at 638272 usecs after Thu Sep
4 16:49:09 2014
    Previous state: [LCM_ST_LC_NOT_PRESENT]
    Triggered event: [LCM_EV_PFM_MODULE_SUP_INSERTED]
    Next state: [LCM_ST_SUPERVISOR_INSERTED]

2) FSM:<ID(257): Slot 1, node 0x0101> Transition at 638480 usecs after Thu Sep
4 16:49:09 2014
    Previous state: [LCM_ST_SUPERVISOR_INSERTED]
    Triggered event: [LCM_EV_START_SUP_INSERTED_SEQUENCE]
    Next state: [LCM_ST_CHECK_INSERT_SEQUENCE]

3) Event:ESQ_START length:38, at 639085 usecs after Thu Sep  4 16:49:09 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    Seq Type:SERIAL

4) Event:ESQ_REQ length:38, at 668947 usecs after Thu Sep  4 16:49:09 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_MIGUTILS_DAEMON(949), Opc:MTS_OPC_LC_INSERTED(1081)

5) FSM:<ID(257): Slot 1, node 0x0101> Transition at 668974 usecs after Thu Sep
4 16:49:09 2014
    Previous state: [LCM_ST_CHECK_INSERT_SEQUENCE]
    Triggered event: [LCM_EV_LC_ONLINE]
    Next state: [LCM_ST_LC_ONLINE]

6) FSM:<ID(257): Slot 1, node 0x0101> Transition at 798999 usecs after Thu Sep
4 16:49:29 2014
    Previous state: [LCM_ST_LC_ONLINE]
    Triggered event: [LCM_EV_PLUGIN_UP]
    Next state: [LCM_ST_LC_ONLINE]

7) Event:ESQ_START length:38, at 799051 usecs after Thu Sep  4 16:49:29 2014
```

```
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    Seq Type:SERIAL

8) Event:ESQ_REQ length:38, at 799288 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_MIGUTILS_DAEMON(949), Opc:MTS_OPC_LC_INSERTED(1081)

9) Event:ESQ_REQ length:38, at 805215 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_PIXM(176), Opc:MTS_OPC_LC_INSERTED(1081)

10) Event:ESQ_REQ length:38, at 811158 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_IFMGR(179), Opc:MTS_OPC_LC_INSERTED(1081)
    RRtoken:0x0000107E

11) Event:ESQ_RSP length:38, at 822258 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_RX] Src:MTS_SAP_IFMGR(179), Opc:MTS_OPC_LC_INSERTED(1081)
    RRtoken:0x0000107E

12) Event:ESQ_REQ length:38, at 822467 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_IFMGR(179), Opc:MTS_OPC_LC_INSERTED(1081)
    RRtoken:0x00001090

13) Event:ESQ_RSP length:38, at 831933 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_RX] Src:MTS_SAP_IFMGR(179), Opc:MTS_OPC_LC_INSERTED(1081)
    RRtoken:0x00001090

14) Event:ESQ_REQ length:38, at 832069 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_PORT_MANAGER_REQ_HIGH(43), Opc:MTS_OPC_LC_INSERTED(10
81)

15) Event:ESQ_REQ length:38, at 832195 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_PORT_MANAGER_REQ_HIGH(43), Opc:MTS_OPC_LC_INSERTED(10
81)

16) FSM:<ID(257): Slot 1, node 0x0101> Transition at 832242 usecs after Thu Sep
 4 16:49:29 2014
    Previous state: [LCM_ST_LC_ONLINE]
    Triggered event: [LCM_EV_LC_ONLINE]
    Next state: [No transition found]
Curr state: [LCM_ST_LC_ONLINE]
```

Example 4-16   show system internal im event-history module command

```
switch# show system internal im event-history module 1
>>>>FSM: <Module NodeID(0x101)> has 13 logged transitions<<<<<

1) FSM:<Module NodeID(0x101)> Transition at 812168 usecs after Thu Sep  4 16:49:
29 2014
    Previous state: [IM_MOD_ST_MODULE_NOT_EXISTENT]
    Triggered event: [IM_MOD_EV_MOD_INSERTED]
    Next state: [IM_MOD_ST_WAIT_CONFIG_FLUSH]

2) FSM:<Module NodeID(0x101)> Transition at 812435 usecs after Thu Sep  4 16:49:
29 2014
    Previous state: [IM_MOD_ST_WAIT_CONFIG_FLUSH]
    Triggered event: [IM_MOD_EV_CONFIG_FLUSH_BYPASSED]
    Next state: [IM_MOD_ST_WAIT_PLATFORM_INIT]

3) Event:ESQ_START length:38, at 812525 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    Seq Type:SERIAL

4) Event:ESQ_REQ length:38, at 812568 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_CRDCFG_SERVER(975), Opc:MTS_OPC_CRDCFG_API_REQ(482)

5) Event:ESQ_REQ length:38, at 813011 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    Seq:IM module internal initialization

6) Event:ESQ_REQ length:38, at 813050 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    Seq:determine steps to skip

7) Event:ESQ_REQ length:38, at 813077 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_VDC_MGR(357), Opc:MTS_OPC_GET_PORT_VDC_MEMB(20483)

8) Event:ESQ_REQ length:38, at 813101 usecs after Thu Sep  4 16:49:29 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_PIXM(176), Opc:MTS_OPC_IM_SHARED_IF_VDC_MEMBERSHIP_UP
DATE(62523)

9) FSM:<Module NodeID(0x101)> Transition at 813470 usecs after Thu Sep  4 16:49:
29 2014
    Previous state: [IM_MOD_ST_WAIT_PLATFORM_INIT]
    Triggered event: [IM_MOD_EV_PLATFORM_INIT_DONE]
    Next state: [IM_MOD_ST_WAIT_P2_MODULE_INSERT]

10) FSM:<Module NodeID(0x101)> Transition at 823156 usecs after Thu Sep  4 16:49
:29 2014
    Previous state: [IM_MOD_ST_WAIT_P2_MODULE_INSERT]
    Triggered event: [IM_MOD_EV_MOD_INSERTED]
    Next state: [FSM_ST_NO_CHANGE]

11) FSM:<Module NodeID(0x101)> Transition at 823251 usecs after Thu Sep  4 16:49
:29 2014
    Previous state: [IM_MOD_ST_WAIT_P2_MODULE_INSERT]
    Triggered event: [IM_MOD_EV_INTERFACE_CREATE]
    Next state: [IM_MOD_ST_WAIT_INTERFACE_CREATE]

12) FSM:<Module NodeID(0x101)> Transition at 823369 usecs after Thu Sep  4 16:49
:29 2014
    Previous state: [IM_MOD_ST_WAIT_INTERFACE_CREATE]
    Triggered event: [IM_MOD_EV_INTERFACE_CREATE_BYPASSED]
    Next state: [IM_MOD_ST_WAIT_INTERFACE_BIND]

13) FSM:<Module NodeID(0x101)> Transition at 823436 usecs after Thu Sep  4 16:49
:29 2014
    Previous state: [IM_MOD_ST_WAIT_INTERFACE_BIND]
    Triggered event: [IM_MOD_EV_INTERFACE_BIND_BYPASSED]
    Next state: [IM_MOD_ST_MODULE_INIT_DONE]
Curr state: [IM_MOD_ST_MODULE_INIT_DONE]
```

Example 4-17 show system internal vmm event-history module command

```
switch# show system internal vmm event-history module 1
>>>>FSM: <ID(257): Module 1> has 8 logged transitions<<<<<
```

```
1) FSM:<ID(257): Module 1> Transition at 950000 usecs after Wed Apr 24 10:02:15
 2013
    Previous state: [VMM_ST_IDLE]
    Triggered event: [VMM_EV_IF_BIND]
    Next state: [VMM_ST_CHECK_INSERT_SEQUENCE]

2) Event:ESQ_START length:38, at 950000 usecs after Wed Apr 24 10:02:15 2013
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    Seq Type:SERIAL

3) Event:ESQ_REQ length:38, at 990000 usecs after Wed Apr 24 10:02:15 2013
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_ETH_PORT_CHANNEL_MGR(378), Opc:MTS_OPC_IM_IF_VDC_BIN
D(62488)
    RRtoken:0x000019F0

4) Event:ESQ_RSP length:38, at 990000 usecs after Wed Apr 24 10:02:15 2013
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_RX] Src:MTS_SAP_ETH_PORT_CHANNEL_MGR(378), Opc:MTS_OPC_IM_IF_VDC_BIN
D(62488)
    RRtoken:0x000019F0

5) Event:ESQ_REQ length:38, at 990000 usecs after Wed Apr 24 10:02:15 2013
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_TEST_ETHPM(175), Opc:MTS_OPC_IM_IF_VDC_BIND(62488)
    RRtoken:0x000019F5

6) Event:ESQ_RSP length:38, at 990000 usecs after Wed Apr 24 10:02:15 2013
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_RX] Src:MTS_SAP_TEST_ETHPM(175), Opc:MTS_OPC_IM_IF_VDC_BIND(62488)
    RRtoken:0x000019F5

7) Event:ESQ_REQ length:38, at 990000 usecs after Wed Apr 24 10:02:15 2013
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    Type: 0

8) FSM:<ID(257): Module 1> Transition at 990000 usecs after Wed Apr 24 10:02:15
 2013
    Previous state: [VMM_ST_CHECK_INSERT_SEQUENCE]
    Triggered event: [VMM_EV_INSERT_SEQ_DONE]
    Next state: [VMM_ST_IDLE]


    Curr state: [VMM_ST_IDLE]
switch#
```

Example 4-18   show system internal ethpm event-history module command

```
switch# show system internal ethpm event-history module 1
>>>>FSM: <Module NodeID(0x101)> has 8 logged transitions<<<<<

1) FSM:<Module NodeID(0x101)> Transition at 754798 usecs after Thu Sep  4 16:49:
35 2014
    Previous state: [ETHPM_MODULE_ST_MODULE_NOT_EXISTENT]
    Triggered event: [ETHPM_MODULE_EV_IF_BIND_CMD]
    Next state: [FSM_ST_NO_CHANGE]

2) FSM:<Module NodeID(0x101)> Transition at 754834 usecs after Thu Sep  4 16:49:
35 2014
    Previous state: [ETHPM_MODULE_ST_MODULE_NOT_EXISTENT]
    Triggered event: [ETHPM_MODULE_EV_SUP_INSERT]
    Next state: [ETHPM_MODULE_ST_AWAIT_SUP_INSERT]
```

```
3) Event:ESQ_START length:38, at 757831 usecs after Thu Sep  4 16:49:35 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    Seq Type:SERIAL

4) Event:ESQ_REQ length:38, at 757864 usecs after Thu Sep  4 16:49:35 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    Seq:SUP_INTERNAL_INIT

5) Event:ESQ_REQ length:38, at 758367 usecs after Thu Sep  4 16:49:35 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_TX] Dst:MTS_SAP_REGISTRY(0), Opc:MTS_OPC_PSSHELPER_PUB_WRITE(28673)

6) Event:ESQ_RSP length:38, at 759300 usecs after Thu Sep  4 16:49:35 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    [E_MTS_RX] Src:MTS_SAP_REGISTRY(0), Opc:MTS_OPC_PSSHELPER_PUB_WRITE(28673)

7) Event:ESQ_REQ length:38, at 759360 usecs after Thu Sep  4 16:49:35 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    Seq:Update_Sup_Module_PSS

8) FSM:<Module NodeID(0x101)> Transition at 759412 usecs after Thu Sep  4 16:49:
35 2014
    Previous state: [ETHPM_MODULE_ST_AWAIT_SUP_INSERT]
    Triggered event: [ETHPM_MODULE_EV_SUP_INSERT_DONE]
    Next state: [ETHPM_MODULE_ST_MODULE_PRESENT]


    Curr state: [ETHPM_MODULE_ST_MODULE_PRESENT]
switch#
```

Example 4-19 show system internal ethpm event-history module command

```
switch# show system internal ethpm event-history module 1

>>>>FSM: <Module NodeID(0x101)> has 8 logged transitions<<<<<

1) FSM:<Module NodeID(0x101)> Transition at 754798 usecs after Thu Sep  4 16:49:
35 2014
    Previous state: [ETHPM_MODULE_ST_MODULE_NOT_EXISTENT]
    Triggered event: [ETHPM_MODULE_EV_IF_BIND_CMD]
    Next state: [FSM_ST_NO_CHANGE]

2) FSM:<Module NodeID(0x101)> Transition at 754834 usecs after Thu Sep  4 16:49:
35 2014
    Previous state: [ETHPM_MODULE_ST_MODULE_NOT_EXISTENT]
    Triggered event: [ETHPM_MODULE_EV_SUP_INSERT]
    Next state: [ETHPM_MODULE_ST_AWAIT_SUP_INSERT]

3) Event:ESQ_START length:38, at 757831 usecs after Thu Sep  4 16:49:35 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    Seq Type:SERIAL

4) Event:ESQ_REQ length:38, at 757864 usecs after Thu Sep  4 16:49:35 2014
    Instance:257, Seq Id:0x1, Ret:SUCCESS
    Seq:SUP_INTERNAL_INIT
```

Example 4-20 vemcmd show version command

```
switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# attach vem 4
switch(vem-attach)# vemcmd show version command
VEM Version: 5.2.1.SK3.2.0.190-0.4.0
VSM Version: 5.2(1)SK3(2.1)
System Version: Linux 3.13.0-34-generic
```

Example 4-21 vemlog show last command

```
switch# vemlog show last 5
Timestamp Entry CPU Mod Lv Message
Mar 17 14:47:30.124446 28768 0 99 4 Warning Could not get LACP Port for LTL 20
Mar 17 14:48:00.123500 28769 0 99 4 Warning Could not get LACP Port for LTL 22
Mar 17 14:48:00.123500 28770 0 99 4 Warning Could not get LACP Port for LTL 21
Mar 17 14:48:00.123500 28771 0 99 4 Warning Could not get LACP Port for LTL 20
Mar 17 14:48:00.248291 28772 6 0 0 Suspending log
```

Example 4-22 vemlog show info command

```
switch# vemlog show info
Enabled: Yes
    Total Entries: 1498
  Wrapped Entries: 0
     Lost Entries: 0
  Skipped Entries: 0
Available Entries: 27594
 Stop After Entry: Not Specified
```

Example 4-23 vemcmd help command

```
switch# vemcmd help
vemcmd help:
show
show version                     Show the VEM and VSM versions
show card                        Show the card's global info
show vsm uptime                  Show the VSM's uptime
show vlan [vlan] cookie <number>
                                 Show the VLAN list (or a given vlan)
show bd [hwbd] cookie <number>   Show the VLAN/BD table
show bd bd-name <bd-name>        Show the VLAN/BD table for the given BD name
show segment [<segment-id>] cookie <number>
                                 Show the BD for the given segment ID
show igmp <vlan> [detail]        Show IGMP status and tables
show acl                         Show ACL ids
show storm stats                 Show Storm Control Debug Stats
show storm-rate ltl <ltl>        Show Storm Control rate
show storm status                Show Storm Control Ltl Status
show qos node [num|cookie] <number>
                                 Show QoS Node info
show acl debug stats             Show ACL debug stats
clear acl debug stats            Clear ACL debug stats
show dpa config vlan brief       Show VLAN Brief
show dpa config vlan vlan_id <vlan-id>
                                 Show VLAN info
show dpa config port-profile brief
                                 Show Port Profile Brief
show dpa config port-profile pp_id <pp-id>
                                 Show Port Profile Info
show qos debug stats             Show QOS debug stats
clear qos debug stats            Clear QOS debug stats
show dr [vlan] cookie <number>   Show the VLAN Designated Receiver
show l2 <vlan>                   Show the L2 table for a given VLAN
show l2 all cookie <number>      Show the L2 table
show l2 bd-name <bd-name>        Show the L2 table for a given BD name
show l2 segment <seg-id>         Show the L2 table for a given Segment ID
show l2-slotwise                 Dump the l2 table slotwise
show l2-num-entries              Show the num of entries in l2 table
show port-old [priv|vsm] cookie <number>
                                 Show the port table
show port-old [priv|vsm] cookie <number>
                                 Show the port table
show port [internal|system|vsm] cookie <number>
                                 Show port information
show port vlans [internal|system|vsm] cookie <number>
                                 Show port vlan information
show port segments cookie <number>
                                 Show port segment information
show port disable-loop-detect [ltl]
                                 Show port disable-loop-detect state
show get-mac bd-name <bd-name>   Show the  get mac table
show port uufb-override          Show port UUFB override states
show port bpduguard              Show port BPDUGUARD states
err_disable port bringup ltl <ltl>
                                 Err_disable port bringup
show port-drops ingress [internal] cookie <number>
                                 Show port drop counters on all ingress stages
show port-drops egress [internal] cookie <number>
                                 Show port drop counters on all egress stages
show port-drops ingress ltl <number>
          Showport drop counters on all ingress stages o
f a ltl
show port-drops egress ltl <number>
          Showport drop counters on all egress stages of
a ltl
showport-drops ltl <number> Showport drop counters on all ingress & egress
stages of a ltl
show dvport [internal] cookie <number>
                                 Show dvport inforamtion
show ltl range cookie <number>   Show ltl range usage
show portdevice cookie <number>
                                 Show the port device types
show pc cookie <number>          Show the port channel table
dump pc                          Show debug dump for port-channel table
show portmac                     Show the port table MAC entries
show port auto-smac-learning     Show auto static mac learning state
show trunk [priv|vsm] cookie <number>
                                 Show the trunk ports in the port table
show bd-trunk                    Show the BD trunk ports in the port table
show stats cookie <number>       Show port stats
show vxlan interfaces            Show the VXLAN Encap Interfaces
show vxlan-encap ltl <ltl>       Show VXLAN Encap Information
show vxlan-encap mac <MAC.MAC.MAC>
                                 Show VXLAN Encap Information
show vxlan-vtep-map              Show VXLAN VTEP VM mapping Information
show vxlan-stats                 Show VXLAN port stats for all ports
show vxlan-stats bd-all          Show VXLAN port stats for all BDs
show vxlan-stats ltl-detail      Show all VXLAN ports stats detail
show vxlan-stats ltl <ltl>       Show VXLAN port stats detail
show vxlan-stats ltl-detail      Show all VXLAN ports stats detail
show vxlan-stats ltl <ltl> bd-all cookie <number>
                                 Show VXLAN port stats for all BDs
show vxlan-stats ltl <ltl> bd-name <bd-name>
                                 Show VXLAN port stats for a BD
show vxlan-stats ltl <ltl> bd-num <bd-num>
                                 Show VXLAN port stats for a BD
show vxlan-vteps                 Show VXLAN VTEPs
show vxlan-vteps bd-name <bd-name>
                                 Show VXLAN VTEPs
```

```
show vxlan threads               Show the VXLAN thread stats
clear vxlan threads              Clear the VXLAN thread stats
show vxlan udp-port              Show the VXLAN UDP port
show packets                     Show port packet stats
show mempool                     Show the memory pool list
show profile                     Show system profile
show pd-port                     Show the platform-dependent (vssnet) port table
show pd-port vlans    Show the platform-dependent (vssnet) port vlan
table
show pd-port-headroom ltl        Show headroom for a port
show span                        Show SPAN/ERSPAN information
show erspan-capability           Show ERSPAN capability information
show heap                        Show the heap list
show acl pinst                   Show ACL policy instances
dump acl policy <acl id>         Dump ACL policy for given acl id
show acl pinst tables            Show ACL policy instances tables
dump pacl entry                  Show PACL entry
show lacp [ltl] cookie <number>
                                 Show the LACP PDU Cache
show netflow monitor             Show NF Monitors
show netflow interface           Show NF Interfaces
show netflow stats               Show NF CLI session stats
show portsec stats               Show the Port Security Stats
show portsec stats vlan <vlan>   Show the Port Security Stats
                                 for a given VLAN
show portsec stats bd-name <bd-name>
                                 Show the Port Security Stats for given BD
show portsec macs <vlan>         Show the Port Security MACs
                                 for a given VLAN
show portsec macs bd-name <bd-name>
                                 Show the Port Security Macs for given BD
show portsec macs all            Show the Port Security Macs
show qos policy [num|cookie] <number>
                                 Show QoS policy info
show qos pinst num|cookie <number>
                                 Show QoS pinst info
dump qos pinst tables num|cookie <number>
                                 Show QoS table pinst info
show qos queue-stats num|cookie <number>
                                 Show QoS queuing stats
clear qos queue-stats num|cookie <number>
                                 Clear QoS queuing stats
show qos queue-rate num|cookie <number>
                                 Show QoS queuing rate stats
dump qos queue-nodes             show QoS queuing nodes
test respool option <option>     Test the resource pool scheduling APIs
show dhcps vlan                  Show DHCP snoop VLANs
show dhcps interfaces            Show DHCP snoop trusted/untrusted intfs
show dhcps binding               Show binding table entry in VEM
show dai vlan                    Show DAI VLANs
show dai interfaces              Show DAI trusted/untrusted intfs
show ipsg interfaces             Show  IPSG intfs
show pinning                     Show Veth pinning
show static pinning config       Show static pinning config for Veths
show dhcps opt82                 Show DHCP option 82 Information
show dhcps stats                 Show DHCP stats
clear dhcps stats                Clear DHCP stats
show dai stats                   Show  DAI stats
clear dai stats <vlan-id>        Clear  DAI stats
show dhcps log level             Show DHCPS log level on this VEM
show dhcps filter-mode           Show DHCPS Filter-Mode
set dhcps log level <level>      Set DHCPS log level on this VEM
show vsd                         Show all installed VSDs
show vsd ports <number>          Show VSD port details
show iscsi pinning               Show iSCSI pinning
show iscsi nics                  Show iSCSI HW capable nics
show ltl-map                     Show Local - Global LTL map
show arp <VLAN>                  Show ARP Cache for a given VLAN
show arp all cookie <number>     Show ARP Cache
show arp bd-name <bd-name>       Show ARP entry for the given BD name
show arp filter                  Show ARP filter entries
show learnt ip                   Show learnts IPs
show learnt ip port [<ltl>]      Show learnt IPs for a LTL
show ip lisp                     Show LISP Config
show ip lisp stats cookie <number> NULL
                                 Show LISP VEM stats
show ip lisp map-cache [local | remote | <eid>] cookie <number>
                                 Display map cache
show ip lisp map-cache-stats [<eid>]
                                 Display map cache stats for remote entries
clear ip lisp stats              Clear LISP VEM stats
clear ip lisp map-cache [<eid>]
                                 Delete map cache entries
show vsn binding [priv|vsm]      Show the VNS Configuration
show vsn config [unused]         Show the VSN Configuration
show vsn interfaces              Show the VSN L3 interfaces
show tracking [ltl] cookie <number>
                                 Show network-state tracking based information
show tracking config             Show network-state tracking config
show channel type                Show Channel type
show sched stats                 Show scheduler statistics
clear sched stats                Clear scheduler and port statistics
show sched errors                Show scheduler errors
show sched debug                 Show scheduler debugs
show dr stats                    Show DR stats
show ids state                   Show Intrusion Detection System (IDS) state
show pd packet stats             Show Platform specific Packet statistics
clear pd packet stats            Clear Platform specific Packet statistics


--- Set commands - be careful! ---

clear portsec mac-address <MAC.MAC.MAC> [bd-name | vlan] [<vlan>|<bdname>]
                                 Clear portsec mac entry
card uuid vmware <string>        Set the host UUID string
card name <string>               Set the host name string
card ip ddd.ddd.ddd.ddd          Set the management IPv4 address
card ipv6 xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
                                 Set the management IPv6 address
card update_level 0-255          Set the ESX Version Update level
set iscsi nic <vmnicN>           Set vmnics that are iSCSI HW capable
clear iscsi nic <vmnicN>         Clear iSCSI HW capability of a vmnic
set iscsi pinning <vmk-ltl> <vmnic-ltl>
                                 Manually pin vmknic to vmnic, overrides auto
set port-mode {trunk|access} [native-vlan <vlan>] ltl <ltl-num>
           Set port mode and native VLAN as system propert
ies
set VNS mac-move-timer <seconds>
                                 Set VNS-VM MAC-refresh timer
set palo-enic <vmnicN>           Set a flag this vmnic is a Palo nic
clear palo-enic <vmnicN>         Clear the flag that vmnic is a Palo nic
show palo-enic                   Show all the Palo Nics on ESX
show pc-internal                 Show PC Internal info.
set ids <enable | disable>       Enable/Disable Intrusion Detection System (IDS)
show dpa heap usage              Display the DPA heap objects
show dp heap usage               Display the DP heap objects
show aclflow stats               Display ACL flow stats
show aclflows [permit|deny]      Display ACL flows
```

```
flush aclflows                   Flush all ACL flows
show acllog config               Show ACL-Log config on VEM
show aclflow dbgstats            Display ACL flow stats
clear aclflow dbgstats           Clear ACL flow debug stats
set card type <vem|vxgw|cgu|cgp>
                                 Set Card Type
show card type                   Get Card Type
show l3sec                       shows l3sec mode and state info
show l3ctrl ipaddr               Get L3-Control IP-Address
show stun rate                   Get STUN allowed rate
show n1kv processes              Show n1kv processes information
show n1kv dp-threads             Show n1kv dp thread information
show ovsswitch                   Print out socket information
get ovsconfig <socket number>    Send OFPT_GET_CONFIG_REQUEST
show ovsports <socket number>    Print out port information
show ofp stats <socket number>   Print out OFP stats
system profile <name> <physical|virtual|access <vlan>|trunk <vlan list>|duplex
                                 Switch profile
profile install                  Install privileged profile <ltl> <profile>
set switch data <file-name>      Set switch opaque data
attach vm <port-uuid> <vm-uuid> <port-number> <vmname> <macaddr> <pgname> [ltl
                                 Attach a VM port to an LTL
detach vm <Port UUID>            Detach a VM port from an LTL
vlan <vlan list>                 Add one or more vlans
no vlan <vlan list>              Remove one or more vlans
port <ifname> <mode> <vlan list>
                                 Enable a port
cbl state <ltl> <vlan> <state>   Set CBL state for a LTL/VLAN
notify ports <physical|virtual|all>
                                 Send notifications for ports
attach pnic <port-uuid> <vm-uuid> <port-number> <portname> <macaddr> <pgname>
                                 Attach a physical port to an LTL
detach pnic <Port UUID>          Detach a physical port from an LTL
switch uuid <switch-uuid>        Set the switch UUID
attach port <port-name> profile <pgname>
                                 Attach Port to a Port-Profile
reread config                    Read and Store config data in config file
offload clear-all                Clear offloading of all flows
ovs-threshold <low> <high>       Set ovsk-ovsd netlink socket thresholds
offload clear <dmac>             Clear offloading of dmac
config lacp-fp enable/disable    Set lacp fastpath to on/off
show vxlan-gw-mappings cookie <number>
                                 Show VXLAN GW VXLAN-VLAN mappings
show vxlan-gw-ha-state           Show VXLAN GW HA State
show offload status [<ltl>]      Show if flow programming is enabled
show flow-mgr status             Show fmgr status
set offload <on | off> [<ltl>]   Enable or disable offload
set offload flow <poll-timeout|rapid-timeout|l2-timeout|stats-timeout> <timeou
                                 Set offload flow timeout
show opaque data                 Display switch opaque data
klm pktdebug enable/disable      Enable/disable klm packet debugging
klm debug enable/disable         Enable/disable klm debugging
show klm                         Display KLM info
show klm flows                   Display KLM flows
show klm l2id [l2id]             Display L2ID Info
show klm port [<ltl>]            Display KLM port info
show klm pc                      Display KLM PC data
show klm port stats [<ltl>]      Display KLM port stats
show klm port stats tab [<ltl>]
                                 Display KLM port stats in tabular form
show klm port stats fp [<ltl>]   Display KLM port stats (fast path pkts only)
show klm port stats tab fp [<ltl>]
           Display KLM port stats (fast path pkts only) in
tabular form
```

```
show klm port stats vss [<ltl>]
                                 Display KLM port stats (vssnet pkts only)
show klm port stats tab vss [<ltl>]
           Display KLM port stats (vssnet pkts only) in ta
bular form
show klm port rates [<interval> [<iterations>]]
                                 Display KLM port rates (summary)
show klm port rates detailed [<interval> [<iterations>]]
                                 Display KLM port rates (detailed)
show klm punt stats              Display KLM punt stats
show klm punt stats tab          Display KLM punt stats in tabular form
show klm punt stats port [<ltl>]
                                 Display KLM punt stats for port
show klm punt stats tab port [<ltl>]
                                 Display KLM punt stats for port in tabular form
show klm punt reasons            Display KLM punt reasons
show klm punt reasons port [<ltl>]
                                 Display KLM punt reasons for port
show klm punt reasons tab port [<ltl>]
           Display KLM punt reasons for port in tabular fo
rm
show klm punt reasons pri        Display KLM punt reasons per priority
showklmpuntreasonstabpri DisplayKLMpuntreasonsperpriorityintabula
r form
clear klm punt stats             Clear KLM punt stats
show klm vxlan source-vteps      Display KLM VXLAN Src VTEPs
show klm ip-mac                  Display KLM IP MAC Binding Table
show klm l2map                   Display KLM VLAN <-> VXLAN Mapping Table
show l2flows                     Display l2flows in user-space
flush l2flows                    Flushes l2flows
show featflows [create]          Display feature flows in user-space
flush featflows                  Flushes feature flows
show klm span [ses_id]           Display KLM SPAN session data
show klm span-source port [ltl]
                                 Display KLM SPAN source port data
show klm span-source vlan [vlan]
                                 Display KLM SPAN source vlan data
show klm span ltl                Display KLM SPAN ltl data
show cts global                  Show cts global config
show cts interfaces              Show cts interface config
show cts ipsgt                   Show cts ipsgt entries
set cts sgt <sgt_val> ltl <ltl-num>
                                 Set CTS SGT on a port
show cts policy                  Show cts policy
show cts access-list             Show cts access-list
set cts trust <0/1> ltl <ltl-num>
                                 Set CTS Trust on a port
set cts enable <0/1> ltl <ltl-num>
                                 Set CTS Enable on a port
set cts role-based sgt <sgt_val> dgt <dgt_val> access-list <access-list-name>
                                 Set CTS SGT policy global
set cts enforcement <0/1> ltl <ltl-num>
                                 Set CTS Enforcement on a port
set cts propagate <0/1> ltl <ltl-num>
                                 Set CTS Propagate on a port
show vlan-vxlan mapping          Show VXLAN-VLAN Port mappings
show multi-mac-capable interfaces
                                 Show multi-mac capable interfaces
show l2-macdistr-num             Show L2 Mac Distribution entries
```

Example 4-24 vem-support.ps1 command

```
switch# vem-support.ps1

Directory: C:\Program Files (x86)\Cisco\Nexus1000V\Support


Mode LastWriteTime Length Name
---- ------------- ------ ----
d---- 3/17/2013 2:51 PM WIN-35-cisco-vem-2013-0317-1451
```

VEM Log Commands

Use the following commands to control the vemlog:

- **vemlog stop**—Stops the log.
- **vemlog clear**—Clears the log.
- **vemlog start** *number-of-entries*—Starts the log and stops it after the specified number of entries.
- **vemlog stop** *number-of-entries*—Stops the log after the next specified number of entries.
- **vemlog resume**—Starts the log, but does not clear the stop value.

CHAPTER 5

Ports

This chapter describes how to identify and resolve problems with ports.

Information About Interface Characteristics

Before a switch can relay frames from one data link to another, you must define the characteristics of the interfaces through which the frames are received and sent. The configured interfaces can be Ethernet (physical) interfaces, virtual Ethernet interfaces, and the management interface (mgmt0).

Each interface has the following:

- Administrative configuration

  The administrative configuration does not change unless you modify it. This configuration has attributes that you can configure in administrative mode.

- Operational state

  The operational state of a specified attribute, such as the interface speed. This state cannot be changed and is read-only. Some values might not be valid when the interface is down (such as the operational speed).

For a complete description of port modes, administrative states, and operational states, see the *Cisco Nexus 1000V for KVM Interface Configuration Guide, Release 5.x.*

Information About Interface Counters

Port counters are used to identify synchronization problems. Counters can show a significant disparity between received and transmitted frames. To display interface counters, enter this command:

show interface ethernet *mod/port* **counters**

See Example 5-10 on page 5-9.

Values stored in counters can be meaningless for a port that has been active for an extended period. Clearing the counters provides a better idea of the actual link behavior at the present time. Create a baseline first by clearing the counters by entering this command:

clear counters interface ethernet *mod/port*

Information About Link Flapping

A port that continually goes up and down is called flapping or a link-flapping port. When a port is flapping, it cycles through the following states, in this order, and then starts over again:

1. Initializing—The link is initializing.
2. Offline—The port is offline.
3. Link failure or not connected—The physical layer is not operational and there is no active device connection.

To troubleshoot link flapping, see Information About Link Flapping, page 5-2.

Port Diagnostic Checklist

Use the following checklist to diagnose the port interface activity.

For more information about port states, see the *Cisco Nexus 1000V for KVM Interface Configuration Guide, Release 5.x*.

Checklist	Example	√
Verify that the module is active by entering the **show module command.**	See Example 5-1 on page 5-6.	
Verify that the ports have been created and the state of the interface by entering the **show interface brief** command.	See Example 5-7 on page 5-8.	

Problems with Ports

This section includes possible causes and solutions for the following symptoms:

- Cannot Enable an Interface, page 5-3
- Port Link Failure or Port Not Connected, page 5-3
- Link Flapping, page 5-3
- Port is ErrDisabled, page 5-4
- Port Troubleshooting Commands, page 5-5

Cannot Enable an Interface

Possible Cause	Solution
A Layer 2 port is not associated with an access VLAN or the VLAN is suspended.	1. Verify that the interface is configured in a VLAN by entering the **show interface brief command.** 2. If not already associated, associate the interface with an access VLAN. 3. Determine the VLAN status by entering the **show vlan brief command.** 4. If the VLAN is not already active, configure the VLAN as active by entering these commands: – **config terminal** – **vlan** *vlan-id* – **state active**

Port Link Failure or Port Not Connected

Possible Cause	Solution
The port connection is bad.	1. Verify the port state by entering the **show system internal ethpm info** command**.** 2. Disable and then enable the port by entering these commands: – **shut** – **no shut** 3. Enter **vemlog show all** on the VEM and collect the output.
The link is stuck in the initialization state or the link is in a point-to-point state.	1. Check for the link failure system message “Link Failure, Not Connected” by entering the **show logging** command. 2. Disable and then enable the port by entering these commands: – **shut** – **no shut** 3. Enter **vemlog show all** on the VEM and collect the output.

Link Flapping

When troubleshooting unexpected link flapping, it is important to have the following information:

- Who initiated the link flap.
- The actual reason for the link being down.

Possible Cause	Solution
The bit rate exceeds the threshold and puts the port into an error-disabled state.	Disable and then enable the port by entering these commands: • **shut** • **no shut** The port should return to the normal state.
A hardware failure or intermittent hardware error causes a packet drop in the switch. A software error causes a packet drop. A control frame is erroneously sent to the device.	An external device might choose to initialize the link again when encountering the error. If so, the exact method of link initialization varies by device. 1. Determine the reason for the link flap as indicated by the MAC driver. 2. Use the debug facilities on the end device to troubleshoot the problem.

Port is ErrDisabled

Possible Cause	Solution
The cable is defective or damaged.	1. Verify the physical cabling. 2. Replace or repair defective cables. 3. Reenable the port by entering these commands: • **shut** • **no shut**
You attempted to add a port to a port channel that was not configured identically and the port is then errdisabled.	1. Display the switch log file and identify the exact configuration error in the list of port state changes by entering the **show logging logfile** command. 2. Correct the error in the configuration and add the port to the port channel. 3. Reenable the port by entering these commands: • **shut** • **no shut**
A VSM application error has occurred.	1. Identify the component that had the error while bringing up the port by entering this command: **show logging log file | grep** *interface_number* See Example 5-6 on page 5-8. 2. Identify the error transition by entering this command: **show system internal ethpm event-history interface** *interface_number* 3. Open a support case and submit the output of the above commands. For more information, see Before Contacting Technical Support, page 16-1.

Port Troubleshooting Commands

You can use the commands in this section to troubleshoot problems related to ports.

Command	Purpose
show module *module-number*	Displays the state of a module. See Example 5-1 on page 5-6.
show svs domain	Displays the domain configuration. See Example 5-2 on page 5-7.
show cdp neighbors	Displays the neighbors connected to an interface. See Example 5-3 on page 5-7.
show system internal ethpm event-history interface interface	Displays information about the internal state transitions of the port. See Example 5-4 on page 5-7.
show logging logfile	Displays logged system messages. See Example 5-5 on page 5-7.
show logging logfile | grep *interface_number*	Displays logged system messages for a specified interface. See Example 5-6 on page 5-8.
show interface brief	Displays a table of interface states. See Example 5-7 on page 5-8.
show interface ethernet *mod/port*	Displays the status of a named interface. See Example 5-8 on page 5-9.
show running-config interface ethernet *mod/port* **expand-port-profile**	Displays the configuration for a named Ethernet interface, including the following: • Administrative state • Speed • Trunk VLAN status • Number of frames sent and received • Transmission errors, including discards, errors, CRCs, and invalid fames. See Example 5-9 on page 5-9.
show interface ethernet *mod/port* **counters**	Displays port counters for identifying synchronization problems. For information about counters, see Information About Interface Counters, page 5-1. See Example 5-10 on page 5-9.
show interface vethernet *number*	Displays the vEthernet interface configuration. See Example 5-11 on page 5-9.
show interface *mod/port* **status**	Displays the status of the named interface.
show interface capabilities	Displays a tabular view of all configured port profiles. See Example 5-12 on page 5-10.

Command	Purpose
show interface virtual attach binding	Displays the virtual port mapping for all vEthernet interfaces. See Example 5-13 on page 5-10.
show system internal ethpm errors	Displays the ethpm error logs. See Example 5-14 on page 5-10.
show system internal ethpm event-history errors	Displays the ethpm event logs. See Example 5-15 on page 5-11.
show system internal ethpm info	Displays the internal data structure information. See Example 5-16 on page 5-11.
show system internal ethpm mem-stats	Displays the ethpm memory allocation statistics.See Example 5-17 on page 5-11.
show system internal ethpm msgs	Displays the ethpm message logs. See Example 5-18 on page 5-13.
show system internal vim errors	Displays VIM error logs. See Example 5-19 on page 5-13.
show system internal vim event-history	Displays various VIM event logs. See Example 5-20 on page 5-13.
show system internal vim info	Displays internal data structure information. See Example 5-21 on page 5-14.
show system internal vim mem-stats	Displays memory allocation statistics of ethpm. See Example 5-22 on page 5-15.
show system internal vim msgs	Displays various message logs of ethpm. See Example 5-23 on page 5-16.
show system internal pktmgr interface brief	Displays a summary of the pktmgr interface status and configuration. See Example 5-24 on page 5-16.
show system internal pktmgr client detail	Displays detailed filter information. See Example 5-25 on page 5-17.

For detailed information about the **show** command output, see the *Cisco Nexus 1000V for KVM Command Reference*.

Example 5-1 show module command

```
vsm-p# show module

Mod  Ports  Module-Type                       Model               Status
---  -----  --------------------------------  ------------------  ------------
1    0      Virtual Supervisor Module         Nexus1000V          active *
4    1022   Virtual Ethernet Module           NA                  ok
5    1022   Virtual Ethernet Module           NA                  ok
6    1022   Virtual Ethernet Module           NA                  ok

Mod  Sw                  Hw
```

```
---  ------------------  ------------------------------------------------
1    5.2(1)SK3(2.0.190)  0.0
4    5.2(1)SK3(2.1)      Linux 3.13.0-34-generic
5    5.2(1)SK3(2.1)      Linux 3.13.0-34-generic
6    5.2(1)SK3(2.1)      Linux 3.13.0-34-generic

Mod  Server-IP        Server-UUID                           Server-Name
---  ---------------  ------------------------------------  --------------------
1    172.27.0.36      NA                                    NA
4    172.27.0.215     6AC6E608-C51D-E211-0010-20304050008D  compute-1
5    172.27.0.218     6AC6E608-C51D-E211-0010-20304050001E  network-2
6    172.27.0.216     6AC6E608-C51D-E211-0010-2030405000AD  compute-2

* this terminal session
```

Example 5-2 show svs domain command

```
switch# show svs domain
SVS domain config:
  Domain id:    942
  Control vlan: 1
  Packet vlan:  1
  Control mode: L3
  L3 control interface: mgmt0
  Status: Config push to Management Server successful.
switch#
```

Example 5-3 show cdp neighbors command

```
switch# show cdp neighbors
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute

Device-ID             Local Intrfce Hldtme Capability  Platform      Port ID
vsm-p(2094532764140613037)
                       mgmt0         141    R B T S   Nexus1000V    control0
```

Example 5-4 show system internal ethpm event-history interface command

```
switch# show system internal ethpm event-history interface e1/7
>>>>FSM: <e1/7> has 86 logged transitions<<<<<
1) FSM:<e1/7> Transition at 647054 usecs after Tue Jan  1 22:44..
    Previous state: [PI_FSM_ST_IF_NOT_INIT]
    Triggered event: [PI_FSM_EV_MODULE_INIT_DONE]
    Next state: [PI_FSM_ST_IF_INIT_EVAL]
2) FSM:<e1/7> Transition at 647114 usecs after Tue Jan  1 22:43..
    Previous state: [PI_FSM_ST_IF_INIT_EVAL]
    Triggered event: [PI_FSM_EV_IE_ERR_DISABLED_CAP_MISMATCH]
    Next state: [PI_FSM_ST_IF_DOWN_STATE]
```

Example 5-5 show logging logfile command

```
switch# show logging logfile
 . . .
2014 Sep  4 16:49:08 vsm-p cdm[2329]: %CDM-5-CDM_APP_REGISTER: CDM main SAP(423)
 registered
2014 Sep  4 16:49:08 vsm-p %USER-2-SYSTEM_MSG: CLIS: loading cmd files begin  -
clis
2014 Sep  4 16:49:09 vsm-p vdc_mgr[2360]: %VDC_MGR-5-VDC_STATE_CHANGE: vdc 1 sta
```

```
te changed to create pending
2014 Sep  4 16:49:09 vsm-p module[2370]: %MODULE-5-ACTIVE_SUP_OK: Supervisor 1 i
s active (serial: T5400449D90)
2014 Sep  4 16:49:09 vsm-p cdm[2329]: %CDM-5-CDM_APP_REGISTER: Fwm SAP(602) regi
stered
2014 Sep  4 16:49:09 vsm-p platform[2290]: %PLATFORM-5-MOD_STATUS: Module 1 curr
ent-status is MOD_STATUS_ONLINE/OK
2014 Sep  4 16:49:10 vsm-p cdm[2329]: %CDM-5-CDM_APP_REGISTER: Aclmgr SAP(351) r
egistered
switch#
```

Example 5-6   show logging logfile | grep interface_number command

```
switch# show logging logfile | grep Vethernet3626
2011 Mar 25 10:56:03 n1k-bl %VIM-5-IF_ATTACHED: Interface Vethernet3626
is attached to Network Adapter 8 of gentoo-pxe-520 on port 193 of module
13 with dvport id 6899
2011 Mar 25 11:10:06 n1k-bl %ETHPORT-2-IF_SEQ_ERROR: Error ("Client data
inconsistency") while communicating with component MTS_SAP_ACLMGR for
opcode MTS_OPC_ETHPM_PORT_PRE_CFG (RID_PORT: Vethernet3626)
2011 Mar 25 11:10:06 n1k-bl %ETHPORT-2-IF_DOWN_ERROR_DISABLED: Interface
Vethernet3626 is down (Error disabled. Reason:Client data inconsistency)
```

Example 5-7   show interface brief command

```
switch# show interface brief
--------------------------------------------------------------------------------
Port     VRF          Status IP Address                            Speed    MTU
--------------------------------------------------------------------------------
mgmt0    --           up     172.27.0.36                           1000     1500

--------------------------------------------------------------------------------
Ethernet     VLAN    Type Mode    Status  Reason                   Speed    Port
Interface                                                                   Ch #
--------------------------------------------------------------------------------
Eth4/1       1       eth  trunk   up      none                     unknown  3
Eth4/2       1       eth  trunk   up      none                     unknown  3
Eth5/1       1       eth  trunk   up      none                     unknown  1
Eth5/2       1       eth  trunk   up      none                     unknown  1
Eth6/1       1       eth  trunk   up      none                     unknown  4
Eth6/2       1       eth  trunk   up      none                     unknown  4
2014 Sep 12 07:18:29 vsm-p %USER-2-SYSTEM_MSG: unknown enum:20000, tid(hex):22a01a0,
comp:41, cid(hex):b0 - vsh

--------------------------------------------------------------------------------
Port-channel VLAN    Type Mode    Status  Reason                 Speed     Proto
Interface
--------------------------------------------------------------------------------
Po1          1       eth  trunk   up      none                    a-20G(D) none
Po3          1       eth  trunk   up      none                    a-20G(D) none
Po4          1       eth  trunk   up      none                    a-20G(D) none

--------------------------------------------------------------------------------
Vethernet    VLAN/    Type Mode    Status  Reason                   MTU  Module
             Segment
--------------------------------------------------------------------------------
Veth6        40       virt access  up      none                     1500 4
Veth7        40       virt access  up      none                     1500 6
Veth9        40       virt access  up      none                     1500 5

--------------------------------------------------------------------------------
Port     VRF          Status IP Address                            Speed    MTU
```

```
--------------------------------------------------------------------------------
control0 --           up     --                                    1000     1500

NOTE : * Denotes ports on modules which are currently offline on VSM
```

Example 5-8 show interface ethernet command

```
switch# show interface e1/14
e1/7 is down (errDisabled)
```

Example 5-9 show running-config interface ethernet mod/port expand-port-profile command

```
switch# show running-config interface ethernet 3/2 expand-port-profile

!Command: show running-config interface Ethernet3/2 expand-port-profile
!Time: Thu Feb 14 17:33:21 2013

version 5.2(1)SK1(1.1)

interface Ethernet3/2
  switchport mode private-vlan trunk promiscuous
  switchport private-vlan trunk allowed vlan 214,224,234,244,254,260,284
  switchport private-vlan trunk allowed vlan add 294,298
  switchport private-vlan mapping trunk 264 10,20,30,40,50
  channel-group auto mode on mac-pinning
  no shutdown

switch#
```

Example 5-10 show interface ethernet counters command

```
switch# show interface eth3/3 counters

--------------------------------------------------------------------------------
Port                                 InOctets                      InUcastPkts
--------------------------------------------------------------------------------
Eth3/3                              167944438                           154350

--------------------------------------------------------------------------------
Port                              InMcastPkts                      InBcastPkts
--------------------------------------------------------------------------------
Eth3/3                                  68452                           298184

--------------------------------------------------------------------------------
Port                                OutOctets                     OutUcastPkts
--------------------------------------------------------------------------------
Eth3/3                                1789120                             8738

--------------------------------------------------------------------------------
Port                             OutMcastPkts                     OutBcastPkts
--------------------------------------------------------------------------------
Eth3/3                                   1461                             3172
```

Example 5-11 show interface vethernet command

```
switch# show interface eth4/1
Ethernet4/1 is up
  Hardware: Ethernet, address: 0025.b520.20dd (bia 0025.b520.20dd)
  Port-Profile is sys-uplink-vpc
  MTU 1550 bytes
  Encapsulation ARPA
  Port mode is trunk
  full-duplex, 20 Gb/s
  5 minute input rate 8560 bits/second, 15 packets/second
  5 minute output rate 32 bits/second, 0 packets/second
  Rx
    27410453 Input Packets 0 Unicast Packets
    2346094 Multicast Packets 25064359 Broadcast Packets
    0 Jumbo Packets
    3129145055 Bytes
```

Example 5-12 show interface capabilities command

```
vsm-p# show interface capabilities
Ethernet4/1
  Model:                 --
  Type (Non SFP):        --
  Speed:                 10,100,1000,10000,auto
  Duplex:                half/full/auto
  Trunk encap. type:     802.1Q
  Channel:               yes
  Broadcast suppression: no
  Flowcontrol:           rx-(none),tx-(none)
  Rate mode:             none
  QOS scheduling:        rx-(none),tx-(none)
  CoS rewrite:           yes
  ToS rewrite:           yes
  SPAN:                  yes
```

Example 5-13 show interface virtual attach binding command

```
switch# show interface virtual attach binding

-------------------------------------------------------------------------------
Port        Bind-Type Hypervisor-Port
-------------------------------------------------------------------------------
Veth6       static    3b657f15-258c-4287-ad29-aa2362f1bba1
Veth7       static    b4d06158-f142-4073-82e9-150fa3918431
Veth9       static    bcfbf059-6452-4433-8a1e-46a2afc9a13c
switch#
```

Example 5-14 show system internal ethpm errors command

```
switch# show system internal ethpm errors
1) Event:E_DEBUG, length:90, at 774936 usecs after Fri Sep 12 07:21:21 2014
    [102] ethpm_cli_if_index_verify(2572): Module not ONLINE for ifindex 0x25008
080-0x25008080

2) Event:E_DEBUG, length:90, at 771697 usecs after Fri Sep 12 07:21:21 2014
    [102] ethpm_cli_if_index_verify(2572): Module not ONLINE for ifindex 0x25008
080-0x25008080

3) Event:E_DEBUG, length:90, at 770939 usecs after Fri Sep 12 07:21:21 2014
    [102] ethpm_cli_if_index_verify(2572): Module not ONLINE for ifindex 0x25008
080-0x25008080

4) Event:E_DEBUG, length:90, at 770165 usecs after Fri Sep 12 07:21:21 2014
    [102] ethpm_cli_if_index_verify(2572): Module not ONLINE for ifindex 0x25008
080-0x25008080

5) Event:E_DEBUG, length:90, at 768445 usecs after Fri Sep 12 07:21:21 2014
    [102] ethpm_cli_if_index_verify(2572): Module not ONLINE for ifindex 0x25008
080-0x25008080
```

```
6) Event:E_DEBUG, length:90, at 767550 usecs after Fri Sep 12 07:21:21 2014
    [102] ethpm_cli_if_index_verify(2572): Module not ONLINE for ifindex 0x25008
080-0x25008080

7) Event:E_DEBUG, length:90, at 766780 usecs after Fri Sep 12 07:21:21 2014
    [102] ethpm_cli_if_index_verify(2572): Module not ONLINE for ifindex 0x25008
080-0x25008080

8) Event:E_DEBUG, length:90, at 764775 usecs after Fri Sep 12 07:21:21 2014
    [102] ethpm_cli_if_index_verify(2572): Module not ONLINE for ifindex 0x25008
080-0x25008080

9) Event:E_DEBUG, length:90, at 762975 usecs after Fri Sep 12 07:21:21 2014
    [102] ethpm_cli_if_index_verify(2572): Module not ONLINE for ifindex 0x25008
080-0x25008080

10) Event:E_DEBUG, length:90, at 761602 usecs after Fri Sep 12 07:21:21 2014
    [102] ethpm_cli_if_index_verify(2572): Module not ONLINE for ifindex 0x25008
080-0x25008080
```

Example 5-15 show system internal ethpm event-history errors command

```
switch# show system internal ethpm event-history errors
1) Event:E_DEBUG, length:59, at 900000 usecs after Mon May 27 16:56:25 2013
    [102] ethpm_shared_port_down_notif(616): seqno = 1 const= 0

2) Event:E_DEBUG, length:59, at 900000 usecs after Mon May 27 16:56:25 2013
    [102] ethpm_shared_port_down_notif(616): seqno = 1 const= 0

3) Event:E_DEBUG, length:59, at 830000 usecs after Mon May 27 16:56:25 2013
    [102] ethpm_shared_port_down_notif(616): seqno = 1 const= 0

4) Event:E_DEBUG, length:59, at 830000 usecs after Mon May 27 16:56:25 2013
    [102] ethpm_shared_port_down_notif(616): seqno = 1 const= 0
```

Example 5-16 show system internal ethpm mem-stats command

```
switch# show system internal ethpm mem-stats
ETHPM Log Buffer info:
[Mon May 27 16:57:58 2013] PORT_FSM_ACTION_INIT         fsm->prev_state:22, eve
nt_id: 65, if_index:0x250080c0 (Ethernet3/4), oper_port_state:0x1, layer:0x2
[Mon May 27 16:57:58 2013] PORT_FSM_ACTION_INIT         fsm->prev_state:22, eve
nt_id: 65, if_index:0x25008140 (Ethernet3/6), oper_port_state:0x1, layer:0x2
[Mon May 27 16:57:58 2013] PORT_FSM_ACTION_INIT         fsm->prev_state:22, eve
nt_id: 65, if_index:0x25008180 (Ethernet3/7), oper_port_state:0x1, layer:0x2
[Mon May 27 16:57:58 2013] PORT_FSM_ACTION_INIT         fsm->prev_state:22, eve
nt_id: 65, if_index:0x250081c0 (Ethernet3/8), oper_port_state:0x1, layer:0x2
```

Example 5-17 show system internal ethpm mem-stats command

```
switch# show system internal ethpm mem-stats
Private Mem stats for UUID : Malloc track Library(103) Max types: 5
--------------------------------------------------------------------------------
Curr alloc: 1812 Curr alloc bytes: 119848(117k)

Private Mem stats for UUID : Non mtrack users(0) Max types: 174
--------------------------------------------------------------------------------
Curr alloc: 558 Curr alloc bytes: 64702(63k)

Private Mem stats for UUID : libsdwrap(115) Max types: 22
--------------------------------------------------------------------------------
```

```
Curr alloc: 54 Curr alloc bytes: 1695892(1656k)

Private Mem stats for UUID : Associative_db library(175) Max types: 14
--------------------------------------------------------------------------------
Curr alloc: 278 Curr alloc bytes: 7892(7k)

Private Mem stats for UUID : Associative_db utils library(174) Max types: 4
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : libfsrv(404) Max types: 12
--------------------------------------------------------------------------------
Curr alloc: 161 Curr alloc bytes: 5100(4k)

Private Mem stats for UUID : Event sequence library(158) Max types: 4
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : FSM Utils(53) Max types: 68
--------------------------------------------------------------------------------
Curr alloc: 411 Curr alloc bytes: 92464(90k)

Private Mem stats for UUID : IM LIB(319) Max types: 34
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Bitlogic Library(517) Max types: 5
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Routing Heap Library(519) Max types: 3
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Routing KSINK (misc. utils) Li(521) Max types: 17
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Routing Hash Table Library(520) Max types: 2
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Routing Library for managing m(522) Max types: 6
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Patricia Trie Library(523) Max types: 3
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Cisco Regex Package(525) Max types: 2
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Routing Queue Library(526) Max types: 2
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Routing SYSLOG Library(527) Max types: 13
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Routing IPC Library(528) Max types: 10
--------------------------------------------------------------------------------
```

```
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Slab Library(529) Max types: 2
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Routing TIMER Library(530) Max types: 4
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : TSP Library(531) Max types: 3
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Routing SYSWRAP Library(534) Max types: 2
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : Test Ethernet Port Manager(306) Max types: 162
--------------------------------------------------------------------------------
Curr alloc: 93 Curr alloc bytes: 169672(165k)

Private Mem stats for UUID : Stats Client Library(1047) Max types: 39
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Curr alloc: 3367 Curr alloc bytes: 2155570 (2105k)
```

Example 5-18 show system internal ethpm msgs command

```
switch# show system internal ethpm msgs
1) Event:E_MTS_RX, length:60, at 431007 usecs after Fri Sep 12 03:45:26 2014
    [NOT] Opc:MTS_OPC_VEM_MGR_MOD_STATE_CHANGE(148531), Id:0X001A6B72, Ret:SUCCE
SS
    Src:0x00000101/744, Dst:0x00000101/0, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x00000000, Sync:UNKNOWN, Payloadsize:100
    Payload:
    0x0000:  00 00 00 02 00 00 00 02 00 00 00 0c 00 00 00 2c

2) Event:E_MTS_RX, length:60, at 430416 usecs after Fri Sep 12 03:45:25 2014
    [NOT] Opc:MTS_OPC_VEM_MGR_MOD_STATE_CHANGE(148531), Id:0X001A6B50, Ret:SUCCE
SS
    Src:0x00000101/744, Dst:0x00000101/0, Flags:None
```

Example 5-19 show system internal vim errors command

```
switch# show system internal vim errors
1) Event:E_DEBUG, length:84, at 351393 usecs after Thu Sep 11 23:42:23 2014
    [102] vim_mod_fsm_ac_process_sync_att_ack(2875): Attach Eth5/2 port state out of sync

2) Event:E_DEBUG, length:84, at 381294 usecs after Thu Sep 11 23:42:22 2014
    [102] vim_mod_fsm_ac_process_sync_att_ack(2875): Attach Eth4/2 port state out of sync

3) Event:E_DEBUG, length:84, at 180587 usecs after Thu Sep 11 22:57:28 2014
    [102] vim_mod_fsm_ac_process_sync_att_ack(2875): Attach Eth5/2 port state out of sync
```

Example 5-20 show system internal vim event-history all command

```
switch# show system internal vim event-history all
>>>>FSM: <VIM Global FSM> has 353 logged transitions<<<<<

1) Event:E_VIM_ACT length:44, at 569244 usecs after Thu Sep  4 16:49:34 2014
```

```
    Slot:0

2) Event:E_PSS_RST length:44, at 570715 usecs after Thu Sep  4 16:49:34 2014
    Type:config Keys:6

3) FSM:<VIM Global FSM> Transition at 571171 usecs after Thu Sep  4 16:49:34 201
4
    Previous state: [VIM_N1K_FSM_ST_IDLE]
    Triggered event: [EV_STATELESS_START]
    Next state: [FSM_ST_NO_CHANGE]

4) FSM:<VIM Global FSM> Transition at 631676 usecs after Thu Sep  4 16:49:34 201
4
    Previous state: [VIM_N1K_FSM_ST_IDLE]
    Triggered event: [VIM_N1K_FSM_EV_CDM_SYNC_CHECK]
    Next state: [FSM_ST_NO_CHANGE]

5) FSM:<VIM Global FSM> Transition at 631798 usecs after Thu Sep  4 16:49:34 201
4
    Previous state: [VIM_N1K_FSM_ST_IDLE]
    Triggered event: [VIM_N1K_FSM_EV_CDM_REG]
    Next state: [VIM_N1K_FSM_ST_WAIT_CDM_REGISTER]

6) FSM:<VIM Global FSM> Transition at 656561 usecs after Thu Sep  4 16:49:34 201
4
    Previous state: [VIM_N1K_FSM_ST_WAIT_CDM_REGISTER]
    Triggered event: [VIM_N1K_FSM_EV_CDM_REG_DONE]
    Next state: [VIM_N1K_FSM_ST_IDLE]

7) Event:E_VIM_RDY length:44, at 678017 usecs after Thu Sep  4 16:49:34 2014
    Veths:3 Status:SUCCES
```

Example 5-21 show system internal vim info command

```
switch# show system internal vim info
auto_setup: true
auto_delete: true
issu_in_progress: false
auto_config_purge: true
veth_retention_time: 300 secs
fsm_state: VIM_N1K_FSM_ST_IDLE
veth_restore_and_launch: completed (3)
vim_ready: true
module 4:
  ports: ETH 32, LVETH 990
  node_addr: 0x00000402
  fsm_state: VIM_MOD_FSM_ST_INSERTED
  srv_license_state: licensed
  num_atts_in_progress: 0
  flags: mod=0x00000040, rt_info=0x0001
  lveth4/1:
    if_index: 0x1b030000
    attached: Veth6
    flags: 0x00000040
    attach_cookie: 0x00000001
    port_state: 0x01
    port_state_reason: 2
  Eth4/1:
    if_index: 0x2500c000
    pp_alias: sys-uplink-vpc (3)
    ds_id: 6bd22a84-b262-4327-8bd0-696109748c6a (7)
    ds_port_uuid: f310a54b-3051-4e5b-8807-ec1a2f592028 (4)
    conn_dev: eth1
    mac: 00:25:b5:20:20:dd
    flags: 0x00000304
    attach_cookie: 0x00000002
    port_state: 0x01
    port_state_reason: 2
    port_duplex: 2
    port_speed: 20000
    auto-negotiate: disabled
  Eth4/2:
    if_index: 0x2500c040
    pp_alias: sys-uplink-vpc (3)
    ds_id: 6bd22a84-b262-4327-8bd0-696109748c6a (7)
    ds_port_uuid: 2e2f9e58-6a6d-4aac-9f25-eeab6c5f9709 (4)
    conn_dev: eth0
    mac: 00:25:b5:20:20:ed
    flags: 0x00000304
    attach_cookie: 0x00000003
    port_state: 0x01
    port_state_reason: 2
    port_duplex: 2
    port_speed: 20000
    auto-negotiate: disabled
  Po3:
```

Example 5-22 show system internal vim mem-stats command

```
switch# show system internal vim mem-stats

Private Mem stats for UUID : Malloc track Library(103) Max types: 5
--------------------------------------------------------------------------------
Curr alloc: 1757 Curr alloc bytes: 117614(114k)

Private Mem stats for UUID : Non mtrack users(0) Max types: 167
--------------------------------------------------------------------------------
Curr alloc: 506 Curr alloc bytes: 53489(52k)

Private Mem stats for UUID : libsdwrap(115) Max types: 22
--------------------------------------------------------------------------------
Curr alloc: 36 Curr alloc bytes: 5092076(4972k)

Private Mem stats for UUID : Associative_db library(175) Max types: 14
--------------------------------------------------------------------------------
Curr alloc: 240 Curr alloc bytes: 6752(6k)

Private Mem stats for UUID : Associative_db utils library(174) Max types: 4
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : libfsrv(404) Max types: 12
--------------------------------------------------------------------------------
Curr alloc: 151 Curr alloc bytes: 4604(4k)

Private Mem stats for UUID : Event sequence library(158) Max types: 4
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)

Private Mem stats for UUID : FSM Utils(53) Max types: 68
--------------------------------------------------------------------------------
Curr alloc: 132 Curr alloc bytes: 71808(70k)

Private Mem stats for UUID : IM LIB(319) Max types: 34
--------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)
```

```
Private Mem stats for UUID : Bitlogic Library(517) Max types: 5
-------------------------------------------------------------------------------
Curr alloc: 0 Curr alloc bytes: 0(0k)
```

Example 5-23 show system internal vim msgs command

```
switch# show system internal vim msgs
1) Event:E_MTS_RX, length:60, at 300185 usecs after Fri Sep 12 07:49:16 2014
    [REQ] Opc:MTS_OPC_SDWRAP_DEBUG_DUMP(1530), Id:0X001B8395, Ret:SUCCESS
    Src:0x00000101/60500, Dst:0x00000101/403, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x001B8395, Sync:UNKNOWN, Payloadsize:216
    Payload:
    0x0000:  01 00 2f 74 6d 70 2f 64 62 67 64 75 6d 70 32 39

2) Event:E_MTS_RX, length:60, at 792564 usecs after Fri Sep 12 07:48:15 2014
    [REQ] Opc:MTS_OPC_SDWRAP_DEBUG_DUMP(1530), Id:0X001B812F, Ret:SUCCESS
    Src:0x00000101/60486, Dst:0x00000101/403, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x001B812F, Sync:UNKNOWN, Payloadsize:208
    Payload:
    0x0000:  01 00 2f 74 6d 70 2f 64 62 67 64 75 6d 70 32 39

3) Event:E_MTS_RX, length:60, at 33426 usecs after Fri Sep 12 07:47:14 2014
    [REQ] Opc:MTS_OPC_VSH_CMD_TLV(7679), Id:0X001B7F8B, Ret:SUCCESS
    Src:0x00000101/60069, Dst:0x00000101/403, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x001B7F8B, Sync:UNKNOWN, Payloadsize:264
    Payload:
    0x0000:  04 03 02 01 08 01 00 00 00 00 00 00 00 00 00 00

4) Event:E_MTS_RX, length:60, at 31581 usecs after Fri Sep 12 07:47:14 2014
    [REQ] Opc:MTS_OPC_VSH_CMD_TLV(7679), Id:0X001B7F89, Ret:SUCCESS
    Src:0x00000101/60069, Dst:0x00000101/403, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x001B7F89, Sync:UNKNOWN, Payloadsize:264
    Payload:
    0x0000:  04 03 02 01 08 01 00 00 00 00 00 00 00 00 00 00

5) Event:E_MTS_RX, length:60, at 28112 usecs after Fri Sep 12 07:47:14 2014
    [REQ] Opc:MTS_OPC_VSH_CMD_TLV(7679), Id:0X001B7F83, Ret:SUCCESS
    Src:0x00000101/60069, Dst:0x00000101/403, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x001B7F83, Sync:UNKNOWN, Payloadsize:244
    Payload:
    0x0000:  04 03 02 01 f4 00 00 00 00 00 00 00 00 00 00 00

6) Event:E_MTS_RX, length:60, at 308316 usecs after Fri Sep 12 07:46:19 2014
    [REQ] Opc:MTS_OPC_SDWRAP_DEBUG_DUMP(1530), Id:0X001B7BE9, Ret:SUCCESS
    Src:0x00000101/60460, Dst:0x00000101/403, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x001B7BE9, Sync:UNKNOWN, Payloadsize:216
    Payload:
    0x0000:  01 00 2f 74 6d 70 2f 64 62 67 64 75 6d 70 32 39

7) Event:E_MTS_RX, length:60, at 597180 usecs after Fri Sep 12 07:45:03 2014
    [REQ] Opc:MTS_OPC_SDWRAP_DEBUG_DUMP(1530), Id:0X001B7A2F, Ret:SUCCESS
```

Example 5-24 show system internal pktmgr interface brief command

```
switch# show system internal pktmgr interface brief
Interface      Type       Interface Status
mgmt0                protocol-up/link-up/admin-up
control0             protocol-up/link-up/admin-up
sup-eth1             protocol-up/link-up/admin-up
sup-eth2             protocol-up/link-up/admin-up
sup-eth3             protocol-up/link-up/admin-up
```

```
port-channel1        protocol-up/link-up/admin-up
port-channel2        protocol-up/link-up/admin-up
```

Example 5-25 show system internal pktmgr client detail command

```
switch# show system internal pktmgr client detail
Client uuid: 268, 3 filters, pid 2422
  Filter 1: EthType 0x0806,
  Rx: 62537, Drop: 0
  Filter 2: EthType 0xfff0, Exc 8,
  Rx: 0, Drop: 0
  Filter 3: EthType 0x8841, Snap 34881,
  Rx: 0, Drop: 0
  Options: TO 0, Flags 0x18040, AppId 0, Epid 0
  Ctrl SAP: 278, Data SAP 337 (1)
  Total Rx: 125074, Drop: 0, Tx: 2906, Drop: 0
  Recirc Rx: 0, Drop: 0
  Rx pps Inst/Max: 0/60
  Tx pps Inst/Max: 0/1
  COS=0 Rx: 0, Tx: 0    COS=1 Rx: 0, Tx: 0
  COS=2 Rx: 0, Tx: 0    COS=3 Rx: 0, Tx: 0
  COS=4 Rx: 0, Tx: 0    COS=5 Rx: 0, Tx: 0
  COS=6 Rx: 0, Tx: 2906    COS=7 Rx: 62537, Tx: 0
```

Port Profiles

This chapter describes how to identify and resolve problems with port profiles.

Information About Port Profiles

Port profiles are used to configure interfaces. A port profile can be assigned to multiple interfaces which gives them all the same configuration. Changes to the port profile are propagated automatically to the configuration of any interface that is assigned to it.

In the KVM Server, a port profile is represented as a port group. The vEthernet or Ethernet interfaces are assigned in the KVM server to a port profile to do the following:

- Define the port configuration by policy.
- Apply a single policy across a large number of ports.
- Support both vEthernet and Ethernet ports.

Port profiles can be assigned by the server administrator to physical ports (a VMNIC or a PNIC). Port profiles that are configured as vEthernet can be assigned only to a vNIC port while port profiles that are configured as Ethernet can be assigned only to physical adapters.

Note While a manual interface configuration overrides that of the port profile, it is not recommended. A manual interface configuration is used only, for example, to quickly test a change or allow a port to be disabled without having to change the inherited port profile.

For more information about assigning port profiles to physical or virtual ports, see *Cisco Nexus 1000V for KVM Port Profile Configuration Guide, Release 5.x.*

To verify that the profiles are assigned as expected to physical or virtual ports, use these **show** commands:

- **show port-profile virtual usage**
- **show running-config interface** *interface-id*

Note You cannot change or remove inherited port profiles from an interface using the Cisco Nexus 1000V CLI. You must use NOVA CLI to detach the interface from the VM and reattach it by creating a new port with new policy profile using the Neutron CLI.

Note When the ports are attached on the hosts, the Cisco Nexus 1000V automatically configures the them with the inherited port profiles. If changes are made to active port profiles, the changes are applied dynamically to each port using the port profile. For detailed information about port profiles, see the *Cisco Nexus 1000V for KVM Interface Configuration Guide, Release 5.x.*

Problems with Port Profiles

The following are symptoms, possible causes, and solutions for problems with port profiles.

Symptom	Possible Causes	Solution
You do not see the port profile/uplink network/network segment on the OpenStack server.	The connection to the OpenStack server is down.	1. Ping the VSM IP. 2. If there is connectivity issue between Neutron and the VSM, fix the network issue.
	The OpenStack server has not pulled the new configuration from the VSM.	Restart the Neutron to ensure that the Neutron configuration is refreshed. Automatic configuration synchronization occurs after the default poll duration (5 minutes by default).
	The port profile is configured incorrectly.	1. To verify that publish port profile is configured for the port profile/Network segment/uplink network, enter this command: **show running-config port-profile** 2. Fix the port profile using the procedures in the *Cisco Nexus 1000V for KVM Port Profile Configuration Guide, Release 5.x.*
A module and all associated interfaces are offline. A system message similar to the following is logged: `2011 Mar 2 22:28:50 n1000v %VEM_MGR-2-VEM_MGR_REMOVE_NO_HB: Removing VEM 3 (heartbeats lost) 2011 Mar 2 22:29:00 n1000v %VEM_MGR-2-MOD_OFFLINE: Module 3 is offline`	The connectivity to the module was lost or the VEM node was powered down.	• Troubleshoot the connectivity to the node hosting the VEM • Power the VEM node if it has shut down.
The interface is in the NoPortProfile state.	The port profile or uplink networks have been deleted from the VSM but are still on the VEM. If the port profiles are used to attach Ethernet and vEthernet interfaces, the interface will go into the NoPortProfile state.	1. Use the saved backup to restore the configuration. 2. Synchronize the Neutron. 3. Copy the running configuration to startup configuration using the **copy run start** command. 4. Reload to apply the configuration to the ports.

Port Profile Logs

To enable and collect detailed logs for port profiles, enter these commands:

- **debug port-profile trace**
- **debug port-profile error**
- **debug port-profile all**
- **debug msp all**
- **debug nsmgr trace**

After enabling the debug log, the results of any subsequent port profile configuration are captured in the log file.

Port Profile Troubleshooting Commands

You can use the commands in this section to troubleshoot problems related to port profiles.

Command	Purpose
show port-profile	Displays the port profile configuration. See Example 6-1 on page 6-4.
show port-profile name *name*	Displays the configuration for a named port profile. See Example 6-1 on page 6-4.
show port-profile brief	Displays a tabular view of all configured port profiles. See Example 6-2 on page 6-4.
show port-profile expand-interface name *name*	Displays a named port profile expanded to include the interfaces assigned to it. See Example 6-3 on page 6-5.
show running-config port-profile [*profile-name*]	Displays the port profile configuration. See Example 6-4 on page 6-5.
show port-profile virtual usage [**name** *profile-name*]	Displays the port profile usage by interface. See Example 6-5 on page 6-6.
show msp internal info	Displays port profile mappings on the KVM server and configured roles. See Example 6-6 on page 6-6.
show system internal port-profile profile-fsm	Displays port profile activity on the Cisco Nexus 1000V, including transitions such as inherits and configurations. If the following appears, all inherits are processed: `Curr state: [PPM_PROFILE_ST_SIDLE]` See Example 6-7 on page 6-7.
show system internal port-profile event-history msgs	Displays the messages logged about port profile events within the Cisco Nexus 1000V. See Example 6-8 on page 6-8.

For detailed information about **show** command output, see the *Cisco Nexus 1000V for KVM Command Reference*.

Examples

Example 6-1 show port-profile name command

```
switch# show port-profile name vEthProfile3
port-profile vEthProfile3
  description:
  type: vethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
    channel-group auto mode on sub-group manual
  evaluated config attributes:
    channel-group auto mode on sub-group manual
  assigned interfaces:
switch#
```

Example 6-2 show port-profile brief command

```
switch# show port-profile brief
--------------------------------------------------------------------------------
Port Profile Profile Conf Eval Assigned Child
Profile Type State Items Items Intfs Profs
--------------------------------------------------------------------------------
LACP Ethernet 1 2 2 0 2
LACP_PIN Ethernet 1 4 5 4 0
MAC Ethernet 1 2 2 0 1
MAC_PIN Ethernet 1 4 5 7 0
MAC_PIN_343 Ethernet 1 2 4 1 0
NSM_template_segmentation Vethernet 1 1 1 0 0
NSM_template_vlan Vethernet 1 1 1 0 0
basic Vethernet 1 1 1 0 0
default Vethernet 1 1 1 0 0
dynpp_a7ab47ce-07c3-4fc8-ae74-321a10818199_76604d2a-f62e-40a4-85d1-0ccad8d1c9c0
Vethernet 1 2 3 0 0
dynpp_a7ab47ce-07c3-4fc8-ae74-321a10818199_aa914386-bf85-48e6-98ca-541a764e7580
Vethernet 1 2 3 2 0
dynpp_a7ab47ce-07c3-4fc8-ae74-321a10818199_b4490e62-57c2-4c3d-81f9-99ca0b6a6a82
Vethernet 1 2 3 8 0
new Vethernet 1 1 1 0 3
system Vethernet 1 1 1 0 0
uplink_network_default_policy Ethernet 1 1 1 0 0
--------------------------------------------------------------------------------
Profile Assigned Total Sys Parent Child UsedBy
Type Intfs Prfls Prfls Prfls Prfls Prfls
--------------------------------------------------------------------------------
Vethernet 10 9 1 8 3 2
Ethernet 12 6 0 4 3 3
switch#
```

Example 6-3 show port-profile expand-interface name UplinkProfile1 command

```
switch# show port-profile expand-interface name UplinkProfile1
port-profile EthProfile1
Ethernet2/2
    switchport mode trunk
    switchport trunk allowed vlan 110-119
    no shutdown
switch#
```

Example 6-4 show running-config port-profile command

```
switch# show running-config port-profile
!Command: show running-config port-profile
!Time: Sun Mar 17 13:17:03 2013

version 5.2(1)SK1(1.1)
port-profile default max-ports 32
port-profile type vethernet NSM_template_vlan
no shutdown
guid 100b8834-85a7-4a9f-a942-83b8218b4fc1
description NSM default port-profile for VLAN networks. Do not delete.
state enabled
port-profile type vethernet NSM_template_segmentation
no shutdown
guid aee2046c-eb9d-4018-bae7-e1000f5b2d54
description NSM default port-profile for VXLAN networks. Do not delete.
state enabled
port-profile type ethernet MAC
channel-group auto mode on mac-pinning
no shutdown
guid 51217cb4-280d-4cbe-a73d-18299cc347c2
max-ports 512
state enabled
port-profile type ethernet LACP
channel-group auto mode active
no shutdown
guid 28a414ca-7c10-4c0d-a73e-a1af409bdb5f
max-ports 512
state enabled
port-profile type vethernet basic
no shutdown
guid bbf3ec9f-9ca3-445a-9376-630180c35250
publish port-profile basic-non-system
state enabled
port-profile type vethernet system
no shutdown
guid 2e21ff4a-e966-4432-95ae-6600e0cbe50f
publish port-profile basic-system
system port-profile
state enabled
port-profile type ethernet uplink_network_default_policy
no shutdown
guid 4cc1067c-7104-4aa1-8556-ce18ada165e8
max-ports 512
description NSM created profile. Do not delete.
state enabled
port-profile type vethernet default
no shutdown
guid 622e109d-6465-4abd-882f-d026938b830d
state enabled
port-profile type vethernet new
no shutdown
```

```
guid a7ab47ce-07c3-4fc8-ae74-321a10818199
publish port-profile
state enabled
switch#
```

Example 6-5 show port-profile virtual usage command

```
switch# show port-profile virtual usage
-------------------------------------------------------------------------------
Port Profile Port Adapter Owner
-------------------------------------------------------------------------------
MAC_PIN Po2
Po6
Eth3/4 vmnic3 WIN-35
Eth3/5 vmnic4 WIN-35
Eth3/6 vmnic5 WIN-35
Eth4/1 vmnic0 WIN-37
Eth4/3 vmnic2 WIN-37
LACP_PIN Po1
Po3
Eth5/1 vmnic0 WIN-39
Eth5/2 vmnic1 WIN-39
dynpp_a7ab47ce-07c3-4fc8-a
e74-321a10818199_b4490e62-
57c2-4c3d-81f9-99ca0b6a6a8
2 Veth1 Net Adapter Win2008-2-1
Veth2 Net Adapter Win2008-1-1
Veth3 Net Adapter Win2008-3-1
Veth4 Net Adapter Win2008-4-1
Veth5 Net Adapter Win2008-2-2
Veth6 Net Adapter Win2008-1-2
Veth7 Net Adapter Win2008-3-2
Veth8 Net Adapter Win2008-4-2
MAC_PIN_343 Po4
dynpp_a7ab47ce-07c3-4fc8-a
e74-321a10818199_aa914386-
bf85-48e6-98ca-541a764e758
0 Veth9 Net Adapter WIN-Legacy
Veth10 Net Adapter WIN-SPAN-3
switch#
```

Example 6-6 show msp internal info command

```
switch# show msp internal info
port-profile NSM_template_segmentation
  id: 2
  capability: 0x0
  state: 0x1
  type: 0x0
  system vlan mode: -
  system vlans:
  port-binding: static
  bind_opts: 0
  max ports: 32
  min ports: 1
  active used ports count: 0
  intf inherit count: 0
  Hyper-V config information
    pg name: NSM_template_segmentation
    dvs:  (ignore)
    reserved ports: 32
  port-profile role:
```

```
  alias information:
    pg id: 8eebad90-fe9a-4460-b44e-9f71b8ebc88d
      dvs uuid:
      type: 11
port-profile NSM_template_vlan
  id: 1
  capability: 0x0
  state: 0x1
  type: 0x0
  system vlan mode: -
  system vlans:
  port-binding: static
  bind_opts: 0
  max ports: 32
  min ports: 1
  active used ports count: 0
  intf inherit count: 0
  Hyper-V config information
    pg name: NSM_template_vlan
    dvs:  (ignore)
    reserved ports: 32
  port-profile role:
  alias information:
    pg id: 83e41305-c443-4d30-a142-f1260183d974
      dvs uuid:
      type: 11
pending binds:
PPM restore_complete:TRUE
  opq_data_info.ppm_sdb_restored:1
NSMGR restore_complete:TRUE
  opq_data_info.nsm_sdb_restored:1
```

Example 6-7 show system internal port-profile profile-fsm command

```
switch# show system internal port-profile profile-fsm
>>>>FSM: <PROFILE_FSM:1> has 4 logged transitions<<<<<

1) FSM:<PROFILE_FSM:1> Transition at 856903 usecs after Tue Mar  8 19:11:47 2011
    Previous state: [PPM_PROFILE_ST_SIDLE]
    Triggered event: [PPM_PROFILE_EV_EIF_STATUS_CHANGE]
    Next state: [PPM_PROFILE_ST_SIDLE]

2) FSM:<PROFILE_FSM:1> Transition at 858442 usecs after Tue Mar  8 19:11:47 2011
    Previous state: [PPM_PROFILE_ST_SIDLE]
    Triggered event: [PPM_PROFILE_EV_ELEARN]
    Next state: [PPM_PROFILE_ST_SIF_CREATE]

3) FSM:<PROFILE_FSM:1> Transition at 842710 usecs after Tue Mar  8 19:12:04 2011
    Previous state: [PPM_PROFILE_ST_SIF_CREATE]
    Triggered event: [PPM_PROFILE_EV_EACKNOWLEDGE]
    Next state: [FSM_ST_NO_CHANGE]

4) FSM:<PROFILE_FSM:1> Transition at 873872 usecs after Tue Mar  8 19:12:04 2011
    Previous state: [PPM_PROFILE_ST_SIF_CREATE]
    Triggered event: [PPM_PROFILE_EV_ESUCCESS]
    Next state: [PPM_PROFILE_ST_SIDLE]

    Curr state: [PPM_PROFILE_ST_SIDLE]
switch#
```

Example 6-8 show system internal port-profile event-history msgs command

```
switch# show system internal port-profile event-history msgs
1) Event:E_MTS_RX, length:60, at 538337 usecs after Tue Mar  8 19:13:02 2011
    [NOT] Opc:MTS_OPC_IM_IF_CREATED(62467), Id:0X0000B814, Ret:SUCCESS
    Src:0x00000101/175, Dst:0x00000101/0, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x00000000, Sync:UNKNOWN, Payloadsize:120
    Payload:
    0x0000:  00 00 00 02 00 00 00 02 00 00 00 0c 00 00 00 29

2) Event:E_MTS_RX, length:60, at 515030 usecs after Tue Mar  8 19:13:02 2011
    [NOT] Opc:MTS_OPC_LC_ONLINE(1084), Id:0X0000B7E8, Ret:SUCCESS
    Src:0x00000101/744, Dst:0x00000101/0, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x00000000, Sync:UNKNOWN, Payloadsize:234
    Payload:
    0x0000:  02 00 00 03 00 00 00 00 00 00 03 02 03 02 00 00

3) Event:E_MTS_RX, length:60, at 624319 usecs after Tue Mar  8 19:12:05 2011
    [NOT] Opc:MTS_OPC_PPM_INTERFACE_UPDATE(152601), Id:0X00003908, Ret:SUCCESS
    Src:0x00000101/489, Dst:0x00000101/0, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x00000000, Sync:UNKNOWN, Payloadsize:107
    Payload:
    0x0000:  00 00 00 02 00 00 00 02 00 00 00 0c 00 00 00 26

4) Event:E_MTS_RX, length:60, at 624180 usecs after Tue Mar  8 19:12:05 2011
    [NOT] Opc:MTS_OPC_PPM_INTERFACE_UPDATE(152601), Id:0X00003905, Ret:SUCCESS
    Src:0x00000101/489, Dst:0x00000101/0, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x00000000, Sync:UNKNOWN, Payloadsize:107
    Payload:
    0x0000:  00 00 00 02 00 00 00 02 00 00 00 0c 00 00 00 26

5) Event:E_MTS_RX, length:60, at 624041 usecs after Tue Mar  8 19:12:05 2011
    [NOT] Opc:MTS_OPC_PPM_INTERFACE_UPDATE(152601), Id:0X00003903, Ret:SUCCESS
    Src:0x00000101/489, Dst:0x00000101/0, Flags:None
    HA_SEQNO:0X00000000, RRtoken:0x00000000, Sync:UNKNOWN, Payloadsize:107
    Payload:
    0x0000:  00 00 00 02 00 00 00 02 00 00 00 0c 00 00 00 26
...
```

CHAPTER 7

Port Channels and Trunking

This chapter describes how to identify and resolve problems that relate to port channels and trunking.

Port Channel Overview

Port channels aggregate multiple physical interfaces into one logical interface to provide higher bandwidth, load balancing, and link redundancy.

A port channel performs the following functions:

- Increases the aggregate bandwidth on a link by distributing traffic among all functional links in the channel.
- Load balances across multiple links and maintains optimum bandwidth usage.
- Provides high availability. If one link fails, traffic previously carried on this link is switched to the remaining links. If a link goes down in a port channel, the upper protocol is not aware of it. To the upper protocol, the link is still there, although the bandwidth is diminished. The MAC address tables are not affected by link failures.

Port Channel Restriction

The following are port channel restrictions:

- Port channels do not support access control lists (ACLs).
- Port channels do not support NetFlow.

Trunking Overview

Trunking, also known as VLAN trunking, enables interconnected ports to transmit and receive frames in more than one VLAN over the same physical link.

Trunking and port channels function as follows:

- Port channels enable several physical links to be combined into one aggregated logical link.
- Trunking enables a link to carry (trunk) multiple VLAN traffic.

Initial Troubleshooting Checklist

Use the following checklist to begin troubleshooting port channel and trunking issues.

Checklist	√
To determine port channel requirements, enter the s**how port-channel compatibility-parameters** command.	
Ensure that all interfaces in the port channel have the same destination device for Link Aggregation Control Protocol (LACP) channels. By using the Asymmetric Port Channel (APC) feature in the Cisco Nexus 1000V, ports in an ON mode channel can be connected to two different destination devices. **Note** APC is supported only in ON mode channels. It is not supported for LACP channels.	
To verify that either side of a port channel is connected to the same number of interfaces, enter these commands: • show port-channel summary • show ether channel summary	
To verify that each interface is connected to the same type of interface on the other side, enter the **show interface brief** command.	
To verify that all required VLANs on a trunk port are in the allowed VLAN list, enter the **show interface switchport** command.	
To verify that all the members trying to form a port channel are on the same module, enter the **show port-channel summary** command.	
To verify that the port channel configuration is present in the profile used by the physical ports, enter the **show port-channel name** *name* command and check for the **channel-group auto** setting.	
Configure APC if the ports are connected to different upstream switches.	
If the upstream switch does not support port channels, make sure to configure APC in the profile.	

The following commands help troubleshoot port channels and trunking:

- **show port-channel summary** (Example 7-1)
- **show port-channel internal event-history interface port-channel** *channel-number*
- **show port-channel internal event-history interface ethernet** *slot-number/port-number*
- **show system internal ethpm event-history interface port-channel** *channel-number*
- **show system internal ethpm event-history interface ethernet** *slot-number/port-number*
- **show vlan internal trunk interface ethernet** *slot-number/port-number*
- **show vlan internal trunk interface port-channel** *channel-number*
- **debug port-channel error**
- **module vem** *module-number* **execute vemcmd show port**
- **module vem** *module-number* **execute vemcmd show port vlans**
- **module vem** *module-number* **execute vemcmd show pc**
- **module vem** *module-number* **execute vemcmd show trunk**

Example 7-1 show port-channel summary Command

```
n1000v# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        S - Switched    R - Routed
        U - Up (port-channel)
        M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port-       Type     Protocol  Member Ports
      Channel
--------------------------------------------------------------------------------
1     Po1(SU)     Eth      NONE      Eth5/1(P)    Eth5/2(P)
3     Po3(SU)     Eth      NONE      Eth4/1(P)    Eth4/2(P)
4     Po4(SU)     Eth      NONE      Eth6/1(P)    Eth6/2(P)

NOTE : * Denotes port-channels on modules that are currently offline on the VSM
```

Verifying a Port Channel Configuration

You can debug port channels that are configured through a port profile.

Step 1 Log in to the CLI in global configuration mode.

Step 2 Verify that you have configured a port channel in the profile by entering the **show port-profile name** *profile-name* command.

Step 3 Verify the port configuration by entering the **show port-channel summary** command.

Step 4 Configure debugging of port-channel trace by entering the **debug port-channel trace** command.

Troubleshooting Asymmetric Port Channels

An asymmetric port channel (APC) is a port channel whose members are connected to two different upstream switches. When troubleshooting asymmetric port channels, follow these guidelines:

- Ports in an APC only come up when they are assigned subgroup IDs.
- For MAC-pinning and MAC-pinning relative to APCs, subgroup IDs are automatically assigned based on the vmnic numbers.
- For the Cisco Discovery Protocol (CDP) subgroup APCs, physical ports within an APC get assigned subgroup IDs based on CDP information received from upstream switches. Make sure that CDP is enabled on the VSM and upstream switches.
- Verify CDP adjacency and subgroup mapping for upstream switches by entering the **show cdp neighbors** and **show port-channel cdp-map** commands on the VSM.
- For manual subgroup APCs, ensure subgroup IDs are manually configured on the physical ports in the interface configuration submode.
- After the ports came up, check that ports are put in the correct subgroups by entering the **module vem** ***module-number*** **execute vemcmd show pc** command on the VSM.
- Configure debugging of port-channel trace by entering the **debug port-channel trace** command.

Troubleshooting LACP Port Channels

The Link Aggregation Control Protocol (LACP) allows you to configure interfaces into a port channel. When troubleshooting LACP port channels, follow these guidelines:

- All physical ports in the port channel should be connected to a single upstream switch.
- The LACP feature should be enabled on both the VSM and the upstream switch.
- The LACP channel group should be configured on all upstream ports and the channel-group ID that is assigned to the upstream ports of a single port channel should be identical.
- At least one end (the Cisco Nexus 1000V or upstream switch) of the port channel should have active LACP mode configured.
- After the ports come up, check that ports are in an LACP port channel by entering the **show lacp port-channel** and **module vem** ***module-number*** **execute vemcmd show pc** commands on the VSM.

Cannot Create a Port Channel

Symptom	Possible Cause	Solution
Cannot create a port channel.	A maximum number of port channels has been reached for the system or VEM.	Verify **the number of port channels already configured by entering** the **show port-channel summary command**. You can have a maximum of 256 port channels on the Cisco Nexus 1000V and 8 port channels per VEM.

Newly Added Interface Does Not Come Online in a Port Channel

Symptom	Possible Cause	Solution
A newly added interface does not come online in a port channel.	The port channel has not been configured.	Make sure that you have the port channel configuration in the port profile (port group) used by that interface.
	The interface parameters are not compatible with the parameters of the existing port.	Configure compatible parameters on all physical ports in a port channel.

VLAN Traffic Does Not Traverse Trunk

Symptom	Possible Cause	Solution
The VLAN traffic does not traverse the trunk.	A VLAN is not in the allowed VLAN list.	Add the VLAN to the allowed VLAN list by entering the **switchport trunk allowed vlan add** *vlan-id* command in the profile that is used by the interface.

CHAPTER 8

Layer 2 Switching

This chapter describes how to identify and resolve problems that relate to Layer 2 switching.

Information About Layer 2 Ethernet Switching

The Cisco Nexus 1000V provides a distributed, Layer 2 virtual switch that extends across many virtualized hosts.

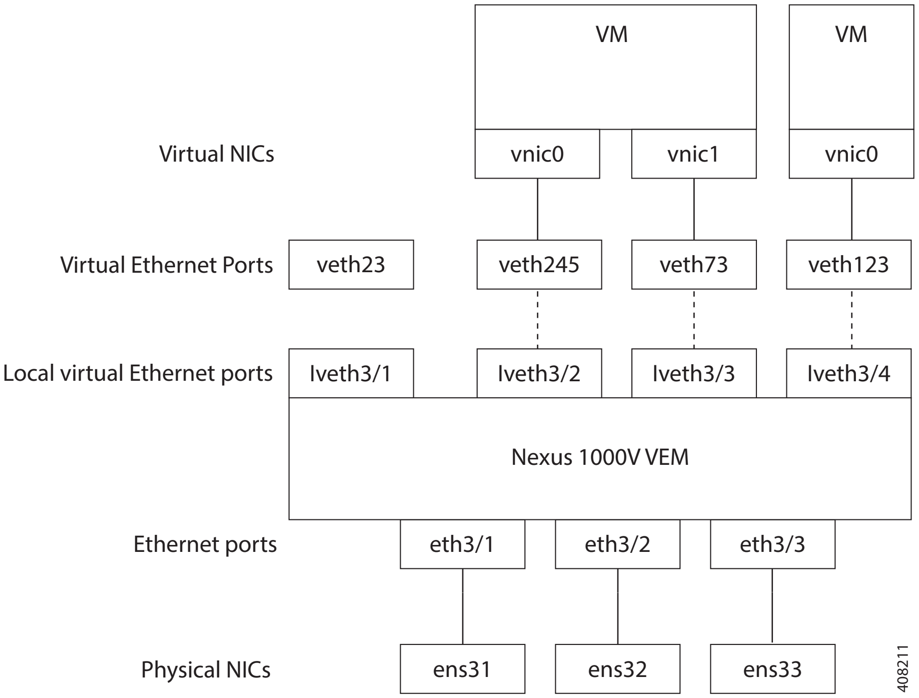
It consists of two components:

- Virtual Supervisor Module (VSM), which is also known as the control plane (CP), acts as the supervisor and contains the Cisco CLI, configuration, and high-level features.
- Virtual Ethernet Module (VEM), which is also known as the data plane (DP), acts as a line card and runs in each virtualized server to handle packet forwarding and other localized functions.

Viewing Ports from the VEM

The Cisco Nexus 1000V differentiates between virtual and physical ports on each of the VEMs. Figure 8-1 shows how ports on the Cisco Nexus 1000V switch are bound to physical and virtual ports within a VEM.

Figure 8-1 VEM View of Ports



On the virtual side of the switch, three layers of ports are mapped together:

- Virtual NICs—There are two types of Virtual NICs. The virtual NIC (vnic) is part of the VM and represents the physical port of the host that is plugged into the switch. Internal NICs are used by the hypervisor for internal purposes. Each type maps to a vEth port within the Cisco Nexus 1000V.
- Virtual Ethernet Ports (VEth)—A vEth port is a port on the Cisco Nexus 1000V distributed virtual switch. The Cisco Nexus 1000V has a flat space of vEth ports 0..N. The virtual cable plugs into these vEth ports that are moved to the host that is running the VM.

  vEth ports are assigned to port groups.
- Local virtual Ethernet ports (lveth)—Each host has a number of local vEth ports. These ports are dynamically selected for vEth ports that are needed on the host.

  These local ports do not move and you can address them by the module-port number method.

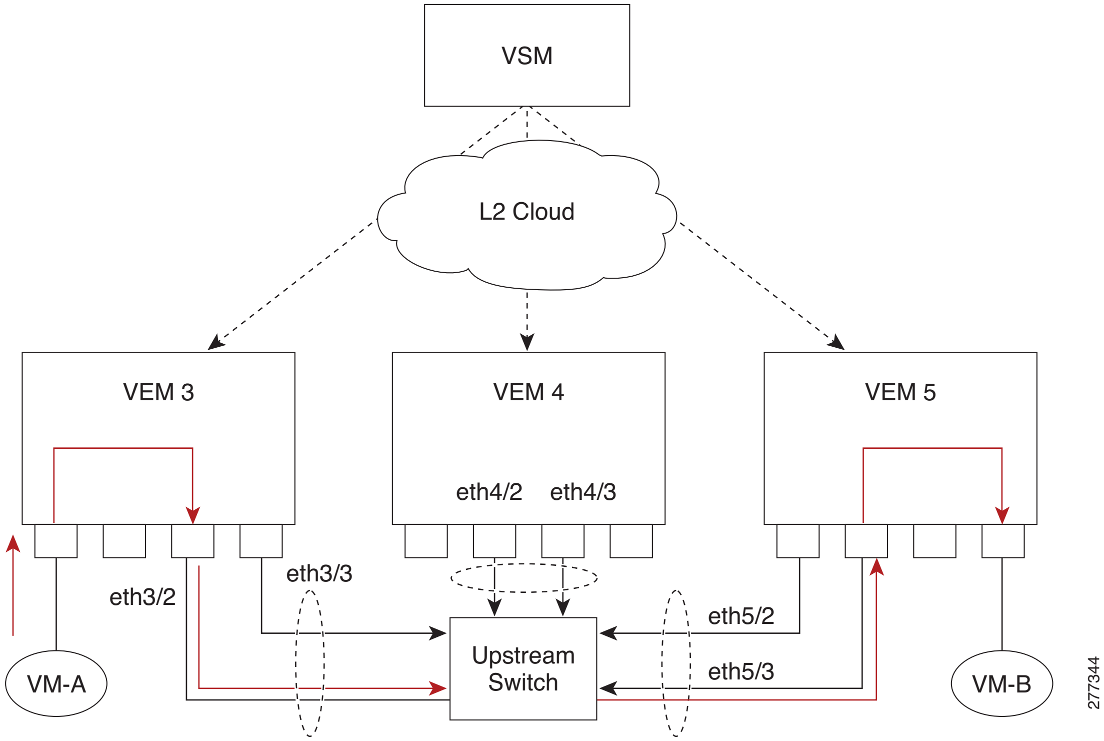
Each physical NIC is represented by an interface. The number is allocated during installation or when a new physical NIC is installed, and remains the same for the life of the host.

Each uplink port on the host represents a physical interface. Each physical port that is added to the Cisco Nexus 1000V switch appears as a physical Ethernet port, just as it would on a hardware-based switch.

Viewing Ports from the VSM

Figure 8-2 shows the VSM view of the ports.

Figure 8-2 VSM View of Ports



Port Types

The following types of ports are available:

- vEths (virtual Ethernet interfaces) can be associated with any one of the following:
  - vNICs of a VM on the hypervisor.
  - Internal NICs on the hypervisor.
- eths (physical Ethernet interfaces)—Correspond to the physical NICs on the hypervisor.
- Po (port channel interfaces)—The physical NICs of a hypervisor can be bundled into a logical interface. This logical bundle is referred to as a port channel interface.

For more information about Layer 2 switching, see the *Cisco Nexus 1000V for KVM Layer 2 Switching Configuration Guide*.

Problems with Layer 2 Switching

This section describes how to troubleshoot Layer 2 problems and lists troubleshooting commands.

Verifying a Connection Between VEM Ports

Step 1 View the state of the VLANs associated with the port by entering the **show vlan** command on the VSM. If the VLAN associated with a port is not active, the port might be down. In this case, you must create the VLAN and activate it.

Step 2 To see the state of the port on the VSM, enter the **show interface brief** command.

Step 3 Display the ports that are present on the VEM, their local interface indices, VLAN, type (physical or virtual), CBL state, port mode, and port name by entering the **module vem** ***module-number*** **execute vemcmd show port** command.

The key things to look for in the output are as follows:

- State of the port.
- CBL.
- Mode.
- Attached device name.
- The LTL of the port that you are trying to troubleshoot. It will help you identify the interface quickly in other VEM commands where the interface name is not displayed.
- Make sure that the state of the port is up. If not, verify the configuration of the port on the VSM.

Step 4 View the VLANs and their port lists on a particular VEM by entering the **module vem** ***module-number*** **execute vemcmd show bd** command.

```
switch# module vem 5 execute vemcmd show bd
```

If you are trying to verify that a port belongs to a particular VLAN, make sure that you see the port name or LTL in the port list of that VLAN.

Verifying a Connection Between VEMs

Step 1 Check if the VLAN associated with the port is created on the VSM by entering the **show vlan** command.

Step 2 Check if the ports are up in the VSM by entering the **show interface brief** command.

Step 3 Check if the CBL state of the two ports is set to the value of 1 for forwarding (active) by entering the **module vem 3 execute vemcmd show port** command on the VEM.

Step 4 Check if the two vEth ports are listed in the flood list of the VLAN to which they are trying to communicate by entering the **module vem 3 execute vemcmd show bd** command on the VEM.

Step 5 Verify that the uplink switch to which the VEMs are connected is carrying the VLAN to which the ports belong.

Step 6 Find the port on the upstream switch to which the physical NIC (that is supposed to be carrying the VLAN) on the VEM is connected to.

```
switch# show cdp neighbors

Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
```

```
Device ID              Local Intrfce   Hldtme Capability  Platform      Port ID
swordfish-6k-2         Eth5/2        168    R S I       WS-C6506-E    Gig1/38
```

The PNIC (Eth 5/2) is connected to swordfish-6k-2 on port Gig1/38.

Step 7 Log in to the upstream switch and make sure the port is configured to allow the VLAN that you are looking for.

```
switch# show running-config interface gigabitEthernet 1/38
Building configuration...

Current configuration : 161 bytes
!
interface GigabitEthernet1/38
 description Srvr-100:vmnic1
 switchport
 switchport trunk allowed vlan 1,60-69,231-233
 switchport mode trunk
end
```

As this output shows, VLANs 1, 60 to 69 and 231 to 233 are allowed on the port. If a particular VLAN is not in the allowed VLAN list, make sure to add it to the allowed VLAN list of the port.

Isolating Traffic Interruptions

Step 1 In the output of the **show port-profile name** command, verify the following information:

- The control and packet VLANs that you configured are present (in the example, these VLANs are 3002 and 3003)
- If the physical NIC in your configuration carries the VLAN for VM, that VLAN is also present in the allowed VLAN list.

```
switch# show port-profile name alluplink
port-profile alluplink
 type: Ethernet
 description:
 status: enabled
 max-ports: 512
 min-ports: 1
 inherit:
 config attributes:
  switchport mode trunk
  switchport trunk allowed vlan 1,80,3002,610,620,630-650
evaluated config attributes:
  switchport mode trunk
  switchport trunk allowed vlan 1,80,3002,3003,610,620,630-650
  no shutdown
assigned interfaces:
    Ethernet2/2
 port-group:
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vn-service: no
 port-profile role: none
 port-binding: static
```

Step 2 Verify that the Ethernet interface is up by entering the **ifconfig –a** command inside the VM.

If not, consider deleting that NIC from the VM, and adding another NIC.

Step 3 Using any sniffer tool, verify that ARP requests and responses are received on the VM interface.

Step 4 On the upstream switch, look for the association between the IP and MAC address by entering these commands:

- **debug arp**
- **show arp**

This example shows how to debug the Address Resolution Protocol (ARP):

```
switch# debug arp
ARP packet debugging is on
11w4d: RARP: Rcvd RARP req for 0050.56b7.3031
11w4d: RARP: Rcvd RARP req for 0050.56b7.3031
11w4d: RARP: Rcvd RARP req for 0050.56b7.4d35
11w4d: RARP: Rcvd RARP req for 0050.56b7.52f4
11w4d: IP ARP: rcvd req src 10.78.1.123 0050.564f.3586, dst 10.78.1.24 Vlan3002
11w4d: RARP: Rcvd RARP req for 0050.56b7.3031
switch#
```

This example shows how to display ARP:

```
switch# show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.78.1.72              -   001a.6464.2008  ARPA
Internet  7.114.1.100             -   0011.bcac.6c00  ARPA   Vlan140
Internet  41.0.0.1                -   0011.bcac.6c00  ARPA   Vlan410
Internet  7.61.5.1                -   0011.bcac.6c00  ARPA   Vlan1161
Internet  10.78.1.5               -   0011.bcac.6c00  ARPA   Vlan3002
Internet  7.70.1.1                -   0011.bcac.6c00  ARPA   Vlan700
Internet  7.70.3.1                -   0011.bcac.6c00  ARPA   Vlan703
Internet  7.70.4.1                -   0011.bcac.6c00  ARPA   Vlan704
Internet  10.78.1.1               0   0011.bc7c.9c0a  ARPA   Vlan3002
Internet  10.78.1.15              0   0050.56b7.52f4  ARPA   Vlan3002
Internet  10.78.1.123            0   0050.564f.3586  ARPA   Vlan3002
```

Layer 2 Switching Troubleshooting Commands

You can use the commands in this section to troubleshoot problems related to the Layer 2 MAC address configuration.

Command	Purpose
show mac address-table	Displays the MAC address table to verify all MAC addresses on all VEMs controlled by the VSM. See Example 8-1 on page 8-7.
show mac address-table module *module-number*	Displays all the MAC addresses on the specified VEM.
show mac address-table static *HHHH.WWWW.HHHH*	Displays the MAC address table static entries. See Example 8-2 on page 8-8.

Command	Purpose
show mac address-table address *HHHH.WWWW.HHHH*	Displays the interface on which the MAC address specified is learned or configured. • For dynamic MAC addresses, if the same MAC address appears on multiple interfaces, then each of them is displayed separately. • For static MAC addresses, if the same MAC address appears on multiple interfaces, then only the entry on the configured interface is displayed.
show mac address-table static | inc veth	Displays the static MAC address of vEthernet interfaces in case a VEM physical port learns a dynamic MAC address and the packet source is in another VEM on the same VSM. See Example 8-3 on page 8-8.
show running-config vlan *vlan-id*	Displays VLAN information in the running configuration.
show vlan [**all-ports** | **brief** | **id** *vlan-id* | **name** *name* | **dot1q tag native**]	Displays VLAN information as specified. See Example 8-4 on page 8-8.
show vlan summary	Displays a summary of VLAN information.
show interface brief	Displays a table of interface states. See Example 8-5 on page 8-9.
module vem *module-number* execute vemcmd show port	On the VEM, displays the port state on a particular VEM. This command can only be used from the VEM. See Example 8-6 on page 8-9.
module vem *module-number* execute vemcmd show bd	For the specified VEM, displays its VLANs and their port lists**.** See Example 8-7 on page 8-9.
module vem *module-number* **execute vemcmd show trunk**	For the specified VEM, displays the VLAN state on a trunk port**.** • If a VLAN is forwarding (active) on a port, its CBL state should be 1. • If a VLAN is blocked, its CBL state is 0. See Example 8-8 on page 8-10.
module vem *module-number* **execute vemcmd show l2** *vlan-id*	For the specified VEM, displays the VLAN forwarding table for a specified VLAN**.** See Example 8-9 on page 8-10.
show interface *interface_id* **mac-address**	Displays the MAC addresses and the burn-in MAC address for an interface.

Example 8-1 show mac address-table command

Note The Cisco Nexus 1000V MAC address table does not display multicast MAC addresses.

Tip Module indicates the VEM on which this MAC address is seen.

The N1KV Internal Port refers to an internal port that is created on the VEM. This port is used for control and management of the VEM and is not used for forwarding packets.

```
switch# show mac address-table
```

```
VLAN      MAC Address       Type    Age       Port                           Mod
---------+-----------------+-------+---------+------------------------------+---
1         0002.3d20.2403    static  0         N1KV Internal Port             4
1         0002.3d30.2403    static  0         N1KV Internal Port             4
1         0002.3d40.2403    static  0         N1KV Internal Port             4
1         0002.3d60.2400    static  0         N1KV Internal Port             4
1         0002.3d80.2403    static  0         N1KV Internal Port             4
1         0000.0c07.accd    dynamic 1         Po3                            4
1         0050.56be.533f    dynamic 136       Po3                            4
1         4403.a74a.8422    dynamic 77        Po3                            4
1         4403.a74a.d586    dynamic 47        Po3                            4
1         5254.000a.ce25    dynamic 5         Po3                            4
1         5254.003e.8614    dynamic 5         Po3                            4
1         5254.0040.9ad6    dynamic 0         Po3                            4

switch#
```

Example 8-2   show mac address-table address command

Tip This command shows all interfaces on which a MAC address is learned dynamically. In this example, the same MAC address appears on Eth3/3 and Eth4/3.

```
switch# show mac address-table address 0050.568d.5a3f
VLAN      MAC Address       Type    Age       Port                           Mod
---------+-----------------+-------+---------+------------------------------+---------
342       0050.568d.5a3f    dynamic 0         Eth3/3                         3
342       0050.568d.5a3f    dynamic 0         Eth4/3                         4
Total MAC Addresses: 1
switch#
```

Example 8-3   show mac address-table static | inc veth command

```
switch# show mac address-table static | inc veth
460       0050.5678.ed16    static  0         Veth2                          3
460       0050.567b.1864    static  0         Veth1                          4
switch#
```

Example 8-4   show vlan command

Tip This command shows the state of each VLAN that is created on the VSM.

```
switch# show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Po1, Po3, Po4, Eth4/1, Eth4/2
                                                Eth5/1, Eth5/2, Eth6/1, Eth6/2
40   VLAN0040                         active    Po1, Po3, Po4, Veth6, Veth7
                                                Veth9, Eth4/1, Eth4/2, Eth5/1
                                                Eth5/2, Eth6/1, Eth6/2

VLAN Type  Vlan-mode
---- ----- ----------
1    enet  CE
40   enet  CE
```

Example 8-5 show interface brief command

```
switch# show interface brief

--------------------------------------------------------------------------------
Port     VRF          Status IP Address                              Speed    MTU
--------------------------------------------------------------------------------
mgmt0    --           up     172.27.0.36                             1000     1500

--------------------------------------------------------------------------------
Ethernet     VLAN    Type Mode    Status  Reason                     Speed    Port
Interface                                                                     Ch #
--------------------------------------------------------------------------------
Eth4/1       1       eth  trunk   up      none                       unknown  3
Eth4/2       1       eth  trunk   up      none                       unknown  3
Eth5/1       1       eth  trunk   up      none                       unknown  1
Eth5/2       1       eth  trunk   up      none                       unknown  1
--More--2014 Sep 12 07:56:34 vsm-p last message repeated 6 times
```

Example 8-6 module vem module-number execute vemcmd show port command

Tip Look for the state of the port.

```
switch# module vem 3 execute vemcmd show port
vsm-p(vem-attach)# vemcmd show port
  LTL   VSM Port  Admin Link  State  PC-LTL  SGID          Vem Port  Type      ORG
svcpath Owner
   18     Eth4/1     UP   UP    FWD    1040     0              eth1              0
0
   19     Eth4/2     UP   UP    FWD    1040     1              eth0              0
0
   50      Veth6     UP   UP    FWD       0     0     cn1-vtep1-ovs  VXLAN       0
0
 1040        Po3     UP   UP    FWD       0                                      0
0

* F/B: Port is BLOCKED on some of the vlans.
       One or more vlans are either not created or
       not in the list of allowed vlans for this port.
 Please run "vemcmd show port vlans" to see the details.
```

Example 8-7 module vem module-number execute vemcmd show bd command

Tip If a port belongs to a particular VLAN, the port name or LTL should be in the port list for the VLAN.

```
switch# module vem 5 execute vemcmd show bd

BD 1, vdc 1, vlan 1, swbd 1, 4 ports, ""

Portlist:
     12  _l24
     18  eth1
     19  eth0
   1040

BD 2, vdc 1, vlan 3972, swbd 3972, 0 ports, ""
```

```
Portlist:
BD 3, vdc 1, vlan 3970, swbd 3970, 0 ports, ""

Portlist:
BD 4, vdc 1, vlan 3968, swbd 3968, 3 ports, ""

Portlist:
      1  inband
      5  inband port security
     11  _l23

BD 5, vdc 1, vlan 3971, swbd 3971, 1 ports, ""

Portlist:
     15  _l27

BD 6, vdc 1, vlan 40, swbd 40, 4 ports, ""

Portlist:
     18  eth1
     19  eth0
     50  cn1-vtep1-ovs
   1040
```

Example 8-8 module vem module-number execute vemcmd show trunk command

Tip If a VLAN is active on a port, its CBL state should be 1. If a VLAN is blocked, its CBL state is 0.

```
switch# module vem 5 execute vemcmd show trunk

Trunk port 6 native_vlan 1 CBL 1
vlan(1) cbl 1, vlan(3972) cbl 1, vlan(3970) cbl 1, vlan(3968) cbl 1, vlan(3971) cbl 1,
vlan(40) cbl 1,
Trunk port 16 native_vlan 1 CBL 1
vlan(1) cbl 1, vlan(3972) cbl 1, vlan(3970) cbl 1, vlan(3968) cbl 1, vlan(3971) cbl 1,
vlan(40) cbl 1,
Trunk port 18 native_vlan 1 CBL 1
vlan(1) cbl 1, vlan(40) cbl 1,
Trunk port 19 native_vlan 1 CBL 1
vlan(1) cbl 1, vlan(40) cbl 1,
Trunk port 1040 native_vlan 1 CBL 1
vlan(1) cbl 1, vlan(40) cbl 1,
```

Example 8-9 module vem module-number execute vemcmd show l2 command

```
switch# configure terminal
n1000v(config)# module vem 3 execute vemcmd show l2
Bridge domain 115 brtmax 1024, brtcnt 2, timeout 300
Dynamic MAC 00:50:56:bb:49:d9 LTL 16 timeout 0
Dynamic MAC 00:02:3d:42:e3:03 LTL 10 timeout 0
switch#
```

VLANs

This chapter describes how to identify and resolve problems that might occur when implementing VLANs.

Information About VLANs

VLANs can isolate devices that are physically connected to the same network but are logically considered to be part of different LANs that do not need to be aware of one another.

We recommend using only the following characters in a VLAN name:

- a-z or A-Z
- 0 to 9
- - (hyphen)
- _ (underscore)

Initial Troubleshooting Checklist

Troubleshooting a VLAN problem involves gathering information about the configuration and connectivity of individual devices and the entire network. In the case of VLANs, begin your troubleshooting activity as follows.

Checklist	√
Verify the physical connectivity for any problem ports or VLANs.	
Verify that both end devices are in the same VLAN.	

The following CLI commands are used to display VLAN information:

- **show system internal private-vlan info**
- **show system internal private-vlan event-history errors**
- **show system internal private-vlan event-history traces**

Cannot Create a VLAN

Symptom	Possible Cause	Solution
You cannot create a VLAN.	• The Cisco Nexus 1000V is using a reserved VLAN ID. • Check the configuration of the type_driver and tenant_network_type in ml2_conf.ini	VLANs 3968 to 4047 and 4094 are reserved for internal use and cannot be changed.

CHAPTER 10

ACLs

This chapter describes how to identify and resolve problems that relate to access control lists (ACLs).

Information About Access Control Lists

An ACL is an ordered set of rules for filtering traffic. When the device determines that an ACL applies to a packet, it tests the packet against the rules. The first matching rule determines whether the packet is permitted or denied. If there is no match, the device applies a default rule. The device processes packets that are permitted and drops packets that are denied.

ACLs protect networks and specific hosts from unnecessary or unwanted traffic. For example, ACLs are used to disallow HTTP traffic from a high-security network to the Internet. ACLs also allow HTTP traffic but only to specific sites, using the IP address of the site to identify it in an IP ACL.

The following types of ACLs are supported for filtering traffic:

- IP ACLs—The device applies IP ACLs only to IP traffic.
- MAC ACLs—The device applies MAC ACLs only to non-IP traffic.

For detailed information about how ACL rules are used to configure network traffic, see the *Cisco Nexus 1000V for KVM Security Configuration Guide, Release 5.x.*

ACL Configuration Limits

The following configuration limits apply to ACLs:

- You cannot have more that 128 rules in an ACL.
- You cannot have more than 128 ACLs (spread across all the ACLs) total in all of the VEMs.

ACL Restrictions

The following restrictions apply to ACLs:

- You cannot apply more than one IP ACL and one MAC ACL in each direction on an interface.
- A MAC ACL applies only to Layer 2 packets.
- VLAN ACLs are not supported.
- ACLs are not supported in port channels.

ACL Troubleshooting Commands

The commands listed in this section can be used on the VSM to see the policies that are configured and applied on the interfaces.

Display the configured ACLs by entering this command:

- **show access-list summary**

Display the run-time information of the ACLMGR and ACLCOMP during configuration errors and to collect ACLMGR process run-time information configuration errors by entering these commands on the VSM:

- **show system internal aclmgr event-history errors**
- **show system internal aclmgr event-history msgs**
- **show system internal aclmgr ppf control**
- **show system internal aclmgr memstat**
- **show system internal aclmgr status**
- **show system internal aclmgr dictionaries**
- **show system internal cdm info app sap 351 detail**

Collect ACLCOMP process run-time information configuration errors by entering these commands:

- **show system internal aclcomp event-history errors**
- **show system internal aclcomp event-history msgs**
- **show system internal aclcomp pdl detailed**
- **show system internal aclcomp mem-stats** (to debug memory usage and leaks)
- **show system internal aclcomp ppf control**

Displaying ACL Policies on the VEM

You can use the commands in this section to display configured ACL policies on the VEM.

To list the ACLs installed on that server, enter this command:

- **module vem** *module-number* **execute vemcmd show acl**

```
n1000v # module vem 3 execute vemcmd show acl
AclId RefCnt Type Rules StatId AclName (Stats: Permit/Deny/NoMatch)
----- ------ ---- ----- ------ ------------------------------------
   1     0   IPv4   1      1   acl-1 (Dis: 0/0/0)
```

The Acl-id is the local ACL ID for this VEM. Ref-cnt refers to the number of instances of this ACL in this VEM.

To list the interfaces on which ACLs have been installed, enter this command:

- **module vem** *module-number* **execute vemcmd show acl pinst**

```
n1000v# module vem 3 execute vemcmd show acl pinst
LTL   Acl-id    Dir
 16       1    ingress
```

Debugging Policy Verification Issues

Step 1 Redirect the output to a file in bootflash by entering the **debug logfile** *filename* command on the VSM.

Step 2 Configure all debug flags of aclmgr by entering the **debug aclmgr all** command.

Step 3 Configure all debug flags of aclcomp by entering the **debug aclcomp all** command.

Step 4 From the VSM enter the following steps:

Note The output goes to the console.

a. Enable ACL logging on the DPA by entering these commands:

   - **module vem** *module-number* **execute vemcmd dpa debug sfaclagent all**

b. Enable logging on the VEM by enter the **module vem** *module-number* **execute vemlog debug sfacl all** command.

c. Enable DPA logging for viewing by entering the **module vem** *module-number* **execute vemlog start** command.

Step 5 Configure the policy that was causing the verification error.

Step 6 Display DPA logs by entering the **module vem** *module-number* **execute vemlog show all** command.

Step 7 Save the Telnet or SSH session buffer to a file.

Step 8 Copy the log file created in bootflash.

CHAPTER 11

Multicast IGMP Snooping

This chapter describes how to identify and resolve problems that relate to multicast Internet Group Management Protocol (IGMP) snooping.

Information About Multicast IGMP Snooping

IP multicast is a method of forwarding the same set of IP packets to a number of hosts within a network. You can use multicast in both IPv4 and IPv6 networks to provide efficient delivery of data to multiple destinations.

Multicast involves both a method of delivery and discovery of senders and receivers of multicast data, which is transmitted on IP multicast addresses called groups. A multicast address that includes a group and source IP address is often referred to as a channel.

IGMP snooping software examines Layer 2 IP multicast traffic within a VLAN to discover the ports where interested receivers reside. Using the port information, IGMP snooping can reduce bandwidth consumption in a multi-access LAN environment to avoid flooding the entire VLAN. The IGMP snooping feature tracks which ports are attached to multicast-capable routers to help the routers forward IGMP membership reports. The IGMP snooping software responds to topology change notifications.

IGMP snooping works as follows:

- Ethernet switches, such as Catalyst 6500 series switches, parse and intercept all IGMP packets and forward them to a CPU, such as a supervisor module, for protocol processing.
- Router ports are learned by using IGMP queries. The switch returns IGMP queries; it remembers which port the query comes from and marks the port as a router port.
- IGMP membership is learned by using IGMP reports. The switch parses IGMP report packets and updates its multicast forwarding table to keep track of IGMP membership.
- When the switch receives multicast traffic, it checks its multicast table and forwards the traffic only to those ports interested in the traffic.
- IGMP queries are flooded to the whole VLAN.
- IGMP reports are forwarded to the uplink port (the router ports).
- Multicast data traffic is forwarded to uplink ports (the router ports).

The operation of multicast IGMP snooping depends on the correct configuration of the upstream switch. Because the IGMP process needs to know which upstream port connects to the router that supports IGMP routing, you must turn on IP multicast routing on the upstream switch by entering the **ip multicast-routing** command.

This example shows how to turn on global multicast routing, configure an SVI interface, and turn on the PIM routing protocol:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip multicast-routing
switch(config)# end

switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# int vlan159
switch(config-if)# ip pim dense-mode
switch(config-if)# end
```

Troubleshooting Guidelines

Follow these guidelines when troubleshooting multicast IGMP issues:

- Verify that IGMP snooping is enabled by entering the **show ip igmp snooping** command.
- Make sure the upstream switch has IGMP configured.
- Verify that the Cisco Nexus 1000V switch is configured correctly and is ready to forward multicast traffic by entering the **show ip igmp snooping groups** command. In the displayed output of the command, look for the letter R under the port heading. The R indicates that the VSM has learned the uplink router port from the IGMP query that was sent by the upstream switch, and means that the Cisco Nexus 1000V is ready to forward multicast traffic.

Multicast IGMP Snooping Troubleshooting Commands

You can use the commands in this section to troubleshoot problems related to multicast IGMP snooping.

Command	Purpose
show cdp neighbor	Displays Cisco Discovery Protocol (CDP) neighbors. However, if you have disabled CDP on the upstream switch by entering the **no cdp enable** command, the **show cdp neighbor** command does not display any information. See Example 11-1 on page 11-2.
show ip igmp snooping groups	Displays if IGMP snooping is enabled on the VLAN. See Example 11-2 on page 11-3.
show ip igmp snooping groups	Displays snooping information for the group addresses.
show ip igmp snooping vlan	Generates additional logs to debug IGMP snooping events on all VLANs. See Example 11-3 on page 11-3.

Example 11-1 show cdp neighbor command

```
n1000V# show cdp neighbor
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
```

```
Device ID              Local Intrfce   Hldtme  Capability  Platform    Port ID
n1000V         Eth3/2           179    R S I       WS-C6506-E    Gig5/16
n1000V         Eth3/4           179    R S I       WS-C6506-E    Gig5/23
```

Example 11-2 show ip igmp snooping vlan command

```
n1000V# show ip igmp snooping vlan 159
IGMP Snooping information for vlan 159
IGMP snooping enabled     <-- IGMP SNOOPING is enabled for vlan 159
Optimised Multicast Flood (OMF) enabled
IGMP querier none
Switch-querier disabled
IGMPv3 Explicit tracking enabled (initializing, time-left: 00:03:20)
IGMPv2 Fast leave disabled
IGMPv1/v2 Report suppression enabled
IGMPv3 Report suppression disabled
Link Local Groups suppression enabled
Router port detection using PIM Hellos, IGMP Queries
Number of router-ports: 0
Number of groups: 0
VLAN vPC function disabled
Active ports:
```

Example 11-3 debug ip igmp snooping vlan command

```
n1000V(config)# debug ip igmp snooping vlan
2008 Sep  2 13:29:36.125661 igmp: SNOOP: <vlan 159> Process a valid IGMP packet
2008 Sep  2 13:29:36.126005 igmp: SNOOP: <vlan 159> Received v2 report: group 224.0.0.251
fro 7.159.159.54 on Vethernet3
2008 Sep  2 13:29:36.126086 igmp: SNOOP: <vlan 159> Added oif Vethernet3 for (*,
224.0.0.251) entry
2008 Sep  2 13:29:36.126157 igmp: SNOOP: <vlan 159> Forwarding report for (*, 224.0.0.251)
came on Vethernet3
2008 Sep  2 13:29:36.126225 igmp: SNOOP: <vlan 159> Forwarding the packet to router-ports
2008 Sep  2 13:29:36.126323 igmp: SNOOP: <vlan 159> Forwarding packet to router-port
Ethernet3/6 (iod 42)
```

On the VSM, use the following command:

- **module vem** *module-number* **execute vemcmd show vlan**

  In Example 11-4, the output shows that LTL 18 corresponds to vmnic3, and LTL 47 corresponds to VM fedora8, interface eth0.

  The multicast group table for 224.1.2.3, shows the interfaces the VEM forwards to when it receives multicast traffic for group 224.1.2.3. If fedora8 has multicast group 224.1.2.3 on its eth0 interface, LTL 47 should be in the multicast group table for 224.1.2.3.

  LTL 18 is also in multicast group 224.1.2.3, which means that it is a VM and generates multicast traffic to 224.1.2.3. The traffic is forwarded to vmnic3, which is the uplink to the upstream switch.

  The multicast group table entry for 0.0.0.0 serves as a default route. If any multicast group traffic does not match any of the multicast groups, the address uses the default route, which means that the traffic is forwarded to an upstream switch through vmnic3.

Example 11-4 module vem module-number execute vemcmd show vlan Command

```
n1000V# module vem 3 execute vemcmd show vlan 159
BD 159, vdc 1, vlan 159, 3 ports
Portlist:
     18  vmnic3
     47  fedora8.eth0
```

```
Multicast Group Table:
Group 224.1.2.3 RID 1 Multicast LTL 4408
    47
    18
Group 0.0.0.0 RID 2 Multicast LTL 4407
    18
```

Problems with Multicast IGMP Snooping

The following are symptoms and solutions for problems with multicast IGMP snooping.

Symptom	Solution
A VM is interested in multicast traffic but is not receiving the multicast traffic	Determine if IGMP snooping is working as expected by entering the **debug ip igmp snooping vlan** command. Examine the output to see if the port is receiving the IGMP report and if the interface has been added to the multicast traffic interface list for the VM.
	Verify that the multicast distribution table in the VEM has the correct information by entering the **module vem** *module-number* **execute vemcmd show vlan** command.
	View the port table by entering the **module vem** *module-number* **execute vemcmd show port** command Make sure that the table has the correct information and that the state of the trunk port and the access port is UP/UP.

CHAPTER 12

Network Segmentation Manager

This chapter describes how to identify and resolve problems with Network Segmentation Manager (NSM).

Information About Network Segmentation Manager

See the *Cisco Nexus 1000V for KVM Virtual Network Configuration Guide, Release 5.x* for more information.

Problems with Network Segmentation Manager

The following are symptoms and possible causes for problems with virtual networking.

Symptom	Solution
OpenStack Neutron operations fails if the VSM is not reachable.	Verify that the VSM and the controller node are connected and resolve connectivity issues, if any.
Creation of networks/subnets/ports fails if it exceeds the quota set in configuration file, neutron.conf	Verify that the quota set in /etc/neutron/neutron.conf is not exceeded. If quota is exceeded, adjust the quota and restart neutron-server on all nodes.
If the VEM port count exceeds 990, vEth ports cannot be created, and this condition impacts data traffic.	Verify whether the number of VEM ports do not exceed 900 virtual ports. Remove some ports, if required.
Neutron agent-list and Nova service-list fails when executing commands.	Verify whether the Neutron and Nova services are correctly configured. Resolve configuration issues, see Red Hat Enterprise Linux OpenStack Platform 7 Director Installation and Usage.

For more information about problems occurring with NSM, see Problems with Port Profiles, page 6-2.

Network Segmentation Manager Troubleshooting Commands

You can use the commands in this section to troubleshoot problems related to the NSM.

Command	Purpose
show nsm ip pool template name ***name***	Displays the IP pool template information.
show nsm ip pool template usage network segment	Displays the network segment using an IP pool template.
show nsm logical network ***name***	Displays the NSM logical network name.
show nsm network segment brief	Displays brief information about the network segment information.
show nsm network segment filter network segment pool *name*	Displays the filtered information for a network segment pool.
show nsm network segment filter vlan *vlan_ID*	Displays the network segment VLAN information.
show nsm network segment name ***name***	Displays network segment information.
show nsm network segment pool ***name***	Displays network segment pool information.
show nsm network uplink brief	Displays brief information about the network segment uplink.
show nsm network uplink filter import *Ethernet Port-Profile name*	Displays network segment uplink information filtered by Ethernet policy port profile.
show dynamic-port-profile	Displays dynamic port profile information.
show dynamic-port-profile ***name***	Displays dynamic port profile information for the specified port profile.
show dynamic-port-profile inherit *name*	Displays dynamic port profiles with inherited vEthernet policy profiles.
show dynamic-port-profile network segment *name*	Displays dynamic port profile network segment information.

For detailed information about **show** command output, see the *Cisco Nexus 1000V for KVM Command Reference, Release 5.x*.

Virtual Networking Troubleshooting Steps

Use the following steps to help you troubleshoot your virtual network:

1. Check the IP address associated with the router.

   # **ip netns exec** *router_namespace_id*

2. Verify that the internal port and external port are connected to br-int.

   # **ovs-vsctl show**

3. Check that the router namespace can ping the router gateway floating IP and fixed IP addresses associated with each instance.

   # **ip netns exec** *router_namespace_id* **ping** [*router_gateway_floating_IP*]

4. Check that you can ping the floating IP and fixed IP addresses that are associated with the instance.

   # **ip netns exec** *router_namespace_id* **ping** *router_gw_IP address*

5. Check up to which port the ping is reachable.

   # **tcpdump -i eth1 | grep ICMP**

6. Check that all internal ports have VLANs configured in the cisco-network-profile-create command.

   # **vemcmd show port vlans**

# **neutron router-list**

# **neutron router-port-list** *router*

7. Verify that the router port is in the VEM and is in the forwarding state.

   # **vemcmd show port**

8. Verify which network node the router is hosted.

   # **neutron l3-agent-list-hosting-router**

   # **neutron router-list-on-l3-agent**

CHAPTER 13

VXLANs

This chapter describes how to identify and resolve problems that might occur when implementing Virtual Extensible Local Area Networks (VXLANs).

Overview

The VXLAN creates LAN segments by using an overlay approach with MAC in IP encapsulation. The encapsulation carries the original Layer 2 (L2) frame from the VM that is encapsulated from within the VEM. Each VEM is assigned an IP address which is used as the source IP address when encapsulating MAC frames to be sent on the network. You can have multiple VXLAN tunnel endpoints (VTEPs) per VEM that are used as sources for this encapsulated traffic. The encapsulation carries the VXLAN identifier which is used to scope the MAC address of the payload frame.

The connected VXLAN is indicated within the port profile configuration of the vNIC and is applied when the VM connects. Each VXLAN uses an assigned IP multicast group to carry broadcast traffic within the VXLAN segment.

When a VM attaches to a VEM, if it is the first to join the particular VXLAN segment on the VEM, an IGMP join is issued for the VXLAN's assigned multicast group. When the VM transmits a packet on the network segment, a lookup is made in the L2 table using the destination MAC of the frame and the VXLAN identifier. If the result is a match, the L2 table entry contains the remote IP address to use to encapsulate the frame and the frame is transmitted within an IP packet destined to the remote IP address. If the result is not a match (broadcast/multicast/unknown unicasts fall into this bucket), the frame is encapsulated with the destination IP address set to be the VXLAN segment's assigned IP multicast group.

When an encapsulated packet is received from the network, it is decapsulated and the source MAC address of the inner frame and VXLAN ID is added to the L2 table as the lookup key and the source IP address of the encapsulation header will be added as the remote IP address for the table entry.

VEM L3 IP Interface for VXLAN

When a VEM has a vEthernet interface connected to a VXLAN, the VEM requires at least one IP/MAC pair to terminate VXLAN packets. In this regard, the VEM acts as an IP host. The VEM only supports IPv4 addressing for this purpose.

Similar to how the VEM Layer 3 (L3) control is configured, the IP address to use for VXLAN is configured by assigning a port profile to a vtep that has the **capability vxlan** command in it.

To support carrying VXLAN traffic over multiple uplinks, or sub-groups, in server configurations where vPC-HM MAC-Pinning is required, up to four vteps with **capability vxlan** may be configured. We recommend that all the VXLAN vteps within the same KVM host are assigned to the same port profile which must have the **capability vxlan** parameter. We can also use the default gateway for different subnets through **transport ip** command.

VXLAN traffic sourced by local vEthernet interfaces is distributed between these vteps based on the source MAC address in their frames. The VEM automatically pins the multiple VXLAN vteps to separate uplinks. If an uplink fails, the VEM automatically repins the vtep to a working uplink.

When encapsulated traffic is destined to a VEM connected to a different subnet, the VEM does not use the VMware host routing table. Instead, the vtep initiates an ARP for the remote VEM IP addresses. The upstream router must be configured to respond by using the Proxy ARP feature.

Fragmentation

The VXLAN encapsulation overhead is 50 bytes. In order to prevent performance degradation due to fragmentation, the entire interconnection infrastructure between all VEMs exchanging VXLAN packets should be configured to carry 50 bytes more than what the VM vNICs are configured to send. For example, using the default vNIC configuration of 1500 bytes, the VEM uplink port profile, upstream physical switch port, and interswitch links, and any routers if present, must be configured to carry an MTU of at least 1550 bytes. If that is not possible, it is suggested that the MTU within the guest VMs be configured to be smaller by 50 bytes, For example, 1450 bytes.

If this is not configured, the VEM attempts to notify the VM if it performs Path MTU (PMTU) Discovery. If the VM does not send packets with a smaller MTU, the VM fragments the IP packets. Fragmentation only occurs at the IP layer. If the VM sends a frame that is too large to carry, after adding the VXLAN encapsulation, and the frame does not contain an IP packet, the frame is dropped.

Scalability

Maximum Number of VXLANs

The Cisco Nexus 1000V supports a total of 4000 VLANs or VXLANs or any combination adding to no more than 2048. This number matches the maximum number of ports on the Cisco Nexus 1000V. Thereby, allowing every port to be connected to a different VLAN or VXLAN.

Supported Features

This section contains the following topics:

- Jumbo Frames, page 13-3
- Disabling the VXLAN Feature Globally, page 13-3

Jumbo Frames

The Cisco Nexus 1000V supports jumbo frames as long as these requirements are met:

- There is room to accommodate the VXLAN encapsulation overhead of at least 50 bytes
- The physical switch or router infrastructure can transport these jumbo sized IP packets.

Disabling the VXLAN Feature Globally

As a safety precaution, the **no feature segmentation** command will not be allowed if there are any ports associated with a VXLAN port profile. You must remove all the associations before disabling the feature. The **no feature segmentation** command will cleanup all the VXLAN Bridge Domain configurations on the Cisco Nexus 1000V.

VXLAN Troubleshooting Commands

Use the following commands to display VXLAN attributes.

VSM Commands

To display ports belonging to a specific segment:

```
switch(config)# show system internal seg_bd info segment 10000
Bridge-domain: A
Port Count: 11
Veth1
Veth2
Veth3
```

To display the vEthernet bridge domain configuration:

```
switch(config)# show system internal seg_bd info port vethernet 1
Bridge-domain: A
segment_id = 10000
Group IP: 225.1.1.1
```

To display the vEthernet bridge configuration with ifindex as an argument:

```
switch(config)# show system internal seg_bd info port ifindex 0x1c000050
Bridge-domain: A
segment_id = 10000
Group IP: 225.1.1.1
```

To display the total number of bridge domain ports:

```
switch(config)# show system internal seg_bd info port_count
Number of ports: 11
```

To display the bridge domain internal configuration:

```
switch(config)# show system internal seg_bd info bd vxlan-home

Bridge-domain vxlan-home (2 ports in all)
Segment ID: 5555 (Manual/Active)
Group IP: 235.5.5.5
State: UP               Mac learning: Enabled
```

```
is_bd_created: Yes
current state: SEG_BD_FSM_ST_READY
pending_delete: 0
port_count: 2
action: 4
hwbd: 28
pa_count: 0
Veth2, Veth5
switch(config)#
```

To display VXLAN vEthernet information:

```
switch# show system internal seg_bd info port
if_index = <0x1c000010>
Bridge-domain vxlan-pepsi
rid = 216172786878513168
swbd = 4098

if_index = <0x1c000040>
Bridge-domain vxlan-pepsi
rid = 216172786878513216
swbd = 4098

switch#
```

Additional **show** commands:

```
show system internal seg_bd info {pss | sdb | global | all}

show system internal seg_bd {event-history | errors | mem-stats | msgs}
```

VEM Commands

To verify VXLAN vEthernet programming:

```
~ # vemcmd show port segments
                          Native  Seg
  LTL   VSM Port  Mode    SegID   State
   50      Veth5   A       5555   FWD
   51      Veth9   A       8888   FWD
~ #
```

To verify VXLAN VTEP programming:

```
~ # vemcmd show vxlan interfaces
LTL           IP       Seconds since Last
                       IGMP Query Received
(* Interface on which IGMP Joins are sent)
------------------------------------------
 49        10.3.3.3        50          *
 52        10.3.3.6        50
~ #
Use “vemcmd show port vlans” to verify that the vteps are in the correct transport VLAN.
```

To verify bridge domain creation on the VEM:

```
~ # vemcmd show bd  bd-name vxlan-home
BD 31, vdc 1, segment id 5555, segment group IP 235.5.5.5, swbd 4098, 1 ports,
"vxlan-home"
Portlist:
     50  RedHat_VM1.eth0

~ #
```

To verify remote IP learning:

```
~ # vemcmd show l2 bd-name vxlan-home
Bridge domain   31 brtmax 4096, brtcnt 2, timeout 300
Segment ID 5555, swbd 4098, "vxlan-home"
Flags:  P - PVLAN  S - Secure  D - Drop
       Type         MAC Address   LTL   timeout   Flags    PVLAN    Remote IP
    Dynamic   00:50:56:ad:71:4e   305         2                     10.3.3.100
     Static   00:50:56:85:01:5b    50         0                      0.0.0.0

~ #
```

To display statistics:

```
~ # vemcmd show vxlan-stats
  LTL  Ucast   Mcast   Ucast   Mcast    Total
       Encaps  Encaps  Decaps  Decaps   Drops
   49       5   14265       4      15       0
   50       6   14261       4      15     213
   51       1      15       0       0      10
   52       0      11       0       0      15

~ #
```

To display detailed per-port statistics for a VXLAN vEthernet/vtep:

```
~ # vemcmd show vxlan-stats ltl 51
```

To display detailed per-port-per-bridge domain statistics for a VXLAN vtep for all bridge domains:

```
~ # vemcmd show vxlan-stats ltl <vxlan_vtep_ltl> bd-all
```

To display detailed per-port-per-bridge domain statistics for a VXLAN vtep for a specified bridge domain:

```
~ # vemcmd show vxlan-stats ltl vxlan_vtep_ltl bd-name bd-name
```

VEM Packet Path Debugging

Use the following commands to debug VXLAN traffic from a VM on VEM1 to a VM on VEM2.

- VEM1: Verify that packets are coming into the switch from the segment vEthernet.

  vempkt capture ingress ltl *vxlan_veth*

- VEM1: Verify VXLAN ecapsulation.

  vemlog debug sflisp all

  vemlog debug sfvnsegment all

- VEM1: Verify that the remote IP address has been learned.

  vemcmd show l2 bd-name *segbdname*

  If the remote IP address has not been learned, then packets are sent as encapsulated multicast packets. For example, an initial ARP request from the VM is sent in this manner.

- VEM1: Find out which uplink is being used and verify that the encapsulated packets are going out the uplink.

  vemcmd show vxlan-encap ltl *ltl*

  vempkt capture egress ltl *uplink*

- VEM1: Display VXLAN statistics and look for any failures.

  vemcmd show vxlan-stats all

  vemcmd show vxlan-stats ltl *veth/vxlanvtep*
- VEM2: Verify that encapsulated packets are arriving on the uplink.

  vempkt capture ingress ltl *uplink*
- VEM2: Verify VXLAN decapsulation.

  vemlog debug sflisp all

  vemlog debug sfvnsegment all
- VEM2: Verify that the decapsulated packets go out on the VXLAN vEthernet port.

  vempkt capture egress ltl *vxlan_veth*
- VEM2: Display VXLAN statistics and look for any failures:

  vemcmd show vxlan-stats all

  vemcmd show vxlan-stats ltl *veth/vxlanvtep*

VEM Multicast Debugging

Use the following command to debug VEM multicast issues.

- IGMP state on the VEM:

```
vemcmd show igmp vxlan_transport_vlan detail
```

Note This command does not show any output for the segment multicast groups. To save multicast table space, segment groups are not tracked by IGMP snooping on the VEM.

- IGMP queries:

Use the **vemcmd show vxlan interfaces** command to verify that IGMP queries are being received.

- IGMP joins from the VTEP:

Use the **vempkt capture ingress ltl** *first_vxlan_vtep_ltl* command to see if the Openstack is sending **join** messages.

Use the **vempkt capture egress ltl** *uplink_ltl* command to see if the **join** messages are being sent to the upstream switch.

VXLAN Datapath Debugging

Use the commands listed in this section to troubleshot VXLAN problems.

Debugging Using the vemlog Command

Command	Result
vemlog debug sfbd all	Displays information to help debug the bridge domain setup or configuration.
vemlog debug sfporttable all	Displays information to help debug the port configuration, CBL, and vEthernet LTL pinning.
vemlog debug sfvnsegment all	Displays information for encapsulation and decapsulation setup and decisions.
vemlog debug sflisp all	Displays information about actual packet editing, VXLAN interface handling, and multicast handling.
echo "debug dpa_allplatform all" > /tmp/dpafifo	Displays multicast joins or leaves on the DPA socket.
echo “debug sfl2agent all” > /tmp/dpafifo	Displays the bridge domain configuration.
echo “debug sfportagent all” > /tmp/dpafifo	Displays debug port configuration information.
echo “debug sfportl2lisp_cache all” > /tmp/dpafifo	Displays debug hitless reconnect (HR) for capability l2-lisp information.
echo “debug sfpixmagent all” > /tmp/dpafifo	Displays debug CBL programming.

HR

To debug segment information for HR, use the following command:

echo “debug sfsegment_cache all” > /tmp/dpafifo (to debug segment info HR)

(now has details of cached and temp segment info list)

echo "show vsm cache *vsm control mac*" **> /tmp/dpafifo**

Vempkt

The **vempkt** command has been enhanced to display VLAN/SegmentID. Use the **vempkt** command to trace the packet path through VEM.

- Encapsulation: Capture ingress on Seg-VEth LTL – Egress on uplink
- Decapsulation: Capture ingress on uplink – Egress on Seg-VEth LTL

Statistics

Command	Result
vemcmd show vxlan-stats	Displays a summary of per-port statistics.
vemcmd show vxlan-stats ltl *vxlan_vtep_ltl*	Displays detailed per-port statistics for VXLAN vtep.
vemcmd show vxlan-stats ltl *vxlan_veth_ltl*	Displays detailed per-port statistics for vEthernet in a VXLAN.

Command	Result
vemcmd show vxlan-stats ltl *vxlan_vtep_ltl* **bd-all**	Displays detailed per-port-per-bridge domain statistics for a VXLAN vtep for all bridge domains.
vemcmd show vxlan-stats ltl *vxlan_vtep_ltl* **bd-name** *bd-name*	Displays detailed per-port-per-bridge domain statistics for a VXLAN vtep for the specified bridge domain.
vemcmd show vxlan-encap ltl *vxlan_veth_ltl*	Displays which VXLAN vtep is used for encap and subsequent pinning to uplink PC for static MAC learned on port.
vemcmd show vxlan-encap mac *vxlan_vm_mac*	Displays which VXLAN vtep is used for encapsulation and subsequent pinning to uplink PC.

Show Commands

Command	Result
vemcmd show vxlan interfaces	Displays the VXLAN encapsulated interfaces.
vemcmd show port vlans	Checks the port programming and CBL state for the bridge domain.
vemcmd show bd	Displays the bridge domain segmentId/group/list of ports.
vemcmd show bd bd-name *bd-name-string*	Displays one segment bridge domain.
vemcmd show l2 all	Displays the remote IP being learned.
vemcmd show l2 bd-name *bd-name-string*	Displays the Layer 2 table for one segment bridge domain.
vemcmd show arp all	Displays the IP-MAC mapping for the outer encapsulated header.

CHAPTER 14

Ethanalyzer

This chapter describes how to use Ethanalyzer as a Cisco NX-OS protocol analyzer tool.

Information About Ethanalyzer

Ethanalyzer is a Cisco NX-OS protocol analyzer tool based on the Wireshark (formerly Ethereal) open source code. Ethanalyzer is a command-line version of Wireshark that captures and decodes packets. You can use Ethanalyzer to troubleshoot your network and analyze the control-plane traffic.

To configure Ethanalyzer, use one or more of the following commands.

Command	Purpose
ethanalyzer local interface *interface*	Captures packets sent or received by the supervisor and provides detailed protocol information. **Note** For all commands in this table, the interface is control, ha-primary, ha-secondary, inband (packet interface) or mgmt (management interface).
ethanalyzer local interface *interface* **limit-captured-frames**	Limits the number of frames to capture.
ethanalyzer local interface *interface* **limit-frame-size**	Limits the length of the frame to capture.
ethanalyzer local interface *interface* **capture-filter**	Filters the types of packets to capture.
ethanalyzer local interface *interface* **display-filter**	Filters the types of captured packets to display.
ethanalyzer local interface *interface* **raw**	Dump the packet in HEX/ASCII with a one line summary.
ethanalyzer local interface *interface* **write**	Saves the captured data to a file.
ethanalyzer local read file	Opens a captured data file and analyzes it.

Ethanalyzer does not capture data traffic that Cisco NX-OS forwards in the hardware. Ethanalyzer uses the same capture filter syntax as tcpdump. For more information, see the following URL:

http://www.tcpdump.org/tcpdump_man.html

For information about the syntax of the display filter, see the following URL:

http://wiki.wireshark.org/DisplayFilters

This example shows captured data (limited to four packets) on the management interface:

```
switch# ethanalyzer local interface mgmt limit-captured-frames 4
Capturing on eth1
2012-10-01 19:15:23.794943 10.78.110.241 -> 72.163.145.51 SSH Encrypted response packet
len=64
2012-10-01 19:15:23.796142 10.78.110.241 -> 72.163.145.51 SSH Encrypted response packet
len=144
2012-10-01 19:15:23.796608 10.78.110.241 -> 72.163.145.51 SSH Encrypted response packet
len=144
2012-10-01 19:15:23.797060 10.78.110.241 -> 72.163.145.51 SSH Encrypted response packet
len=144
4 packets captured
switch#
```

For information about Wireshark, see the following URL: http://www.wireshark.org/docs/

CHAPTER 15

Troubleshooting the Cisco Nexus 1000V Installation

This chapter describes how to identify and resolve problems related to installing the Cisco Nexus 1000V Switch for KVM on the Red Hat Enterprise Linux OpenStack Platform 7 (RHEL-OSP7).

Information About Cisco Nexus 1000V for KVM on the RHEL-OSP

The Cisco Nexus 1000V for KVM on the RHEL-OSP consists of these main components:

- Virtual Ethernet Module (VEM)—A software component that is deployed on each KVM host. Each VM on the host is connected to the VEM through virtual Ethernet (vEth) ports. The VEM is a hypervisor-resident component and is tightly integrated with the KVM architecture.
- Virtual Supervisor Module (VSM)—The management component that controls multiple VEMs and helps in the definition of VM-focused network policies. It is deployed either as a virtual appliance on any KVM host or on the Cisco Cloud Services Platform appliance. The VSM is integrated with OpenStack using the OpenStack Neutron plug-in.

Note This guide does not cover Cisco Nexus 1000V switch installation on the Cloud Services Platform.

- RHEL-OSP—Red Hat Enterprise Linux operating system with the Red Hat implementation of OpenStack Kilo. RHEL-OSP consists of services to control and manage computing, storage, and networking resources. These services provides the foundation to build a private or public Infrastructure-as-a-Service (IaaS) cloud.

The Cisco Nexus 1000V for KVM uses Red Hat's deployment management tool called Red Hat Enterprise Linux OpenStack Platform Director (also known as RHEL-OSPD) to install the Cisco Nexus 1000V for KVM on RHEL in an OpenStack cloud environment. The RHEL-OSP Director is based on the OpenStack-on-OpenStack (TripleO) project. The RHEL-OSP Director consists of two main components:

- Undercloud: The main director node that contains components for configuring and managing the OpenStack nodes that comprise the OpenStack environment (Overcloud). The main components of Undercloud provide functionality for environment planning, bare metal system control, and orchestration for OpenStack environment. For more information on Undercloud, see Red Hat Enterprise Linux OpenStack Platform 7 Director Installation and Usage.

- Overcloud: The RHEL-OSP environment that is created using the Undercloud. The Overcloud comprises three main node types: controller nodes, compute nodes, and storage nodes. For more information on Overcloud, see Red Hat Enterprise Linux OpenStack Platform 7 Director Installation and Usage.

Problems with Cisco Nexus 1000V Installation on OSP7

The following are symptoms, possible causes, and solutions for installation problems.

Symptom	Possible Causes	Solution
Overcloud fails to deploy due to a puppet failure.	The VSM management interface (N1000vVSMHostMgmtIntf) is set to the same interface as the provisioning interface.	Reconfigure the VSM to use a different management interface or refer to the Red Hat documentation and configure the provisioning interface on a bridge. Use the bridge for N1000vVSMHostMgmtIntf and set the N1000vExistingBridge parameter to true. These parameters are defined in the cisco-n1kv-config.yaml configuration file available at */usr/share/openstack-tripleo-heat-templates/environments*.
Some VEMs cannot communicate with the VSM.	• Different vendor NICs are attached to your physical server. The management interface might be mapped to a different Ethernet interface other than the eth0 interface. • The parameters N1000vVSMHostMgmtIntf, N1000vVEMHostMgmtIntf, and N1000vExistingBridge are configured incorrectly in the configuration file, */usr/share/openstack-tripleo-heat-templates/environments/cisco-n1kv-config.yaml*. • The parameter N1000vVEMHostMgmtIntf is configured incorrectly in the environment file.	Edit the parameter values to match the values in the configuration file (YAML file), or the environment file. If your compute and controller nodes have heterogeneous NIC ordering, you can leverage the custom configuration provided by the NodeDataLookup parameter to specify the different configurations. We recommend that you use the NodeDataLookup parameter to specify the configuration for controller nodes and use the N1000vVEMHostMgmtIntf parameter for the compute nodes. The number of controller nodes ranges from 1 to 3, whereas the number of computes nodes can expand over the life of your deployment. For details about how to leverage NodeDataLookup, see the *Cisco Nexus 1000V for KVM Installation Guide for Red Hat Enterprise Linux OpenStack Platform 7*.

Symptom	Possible Causes	Solution
VSM bringup fails if the management (provisioning) interface of the controller node is not eth0.	• Different vendor NICs are attached to your physical server. The management interface might be mapped to a different Ethernet interface other than the eth0 interface. • The parameters N1000vVSMHostMgmtIntf and N1000vExistingBridge are configured incorrectly in the configuration file, */usr/share/openstack-tripleo-heat-templates/environments/cisco-n1kv-config.yaml*.	1. Configure the parameters in the configuration file */usr/share/openstack-tripleo-heat-templates/environments/cisco-n1kv-config.yaml*. 2. If the management interface is not eth0, configure the management interface of VSM on a separate uplink interface. Set the value of the N1000vVSMHostMgmtIntf parameter to an uplink interface name other than the name used for the controller node management interface. Also, set the N1000vExistingBridge parameter to *false*.
VSM boots to the loader prompt after multiple controllers nodes reboot ungracefully.	Multiple controller nodes fail simultaneously.	Ensure that you have a backup of the latest VSM configuration at a remote location. 1. Disable pacemaker resources, such as the primary VSM (vsm-p) and the secondary VSM (vsm-s). 2. Log in to nodes with active VSMs and shut down the VSM VMs. 3. Log in to all three controllers and format the primary_disk and secondary_disk by using the **qemu-img create** *disk-name* **4G** command at */var/spool/cisco/vsm/.* 4. Enable both the primary and secondary VSMs in pacemaker. `#pcs resource enable resource_id` 5. Recover the missing VSM configuration from backup. **Note** The VSM configuration might be lost during the recovery.

Symptom	Possible Causes	Solution

VSMs go into a split-brain condition where the primary and secondary VSMs are in active - active state.	• Layer 2 connectivity between the primary and secondary VSMs is lost.
	• Multiple controller nodes fail simultaneously.
		Ensure that you have a backup of the latest VSM configuration at a remote location.
		1. Identify the primary and secondary VSM controller hosts by using the **pcs status** command. For example:
		`[root@overcloud-controller-2`
		`heat-admin]# pcs status | grep vsm`
		`vsm-p  (ocf::heartbeat:VirtualDomain):`
		`Started overcloud-controller-1`
		`vsm-s  (ocf::heartbeat:VirtualDomain):`
		`Started overcloud-controller-2`
		2. Disable pacemaker resources, such as the primary VSM (vsm-p) and the secondary VSM (vsm-s). For example:
		`#pcs resource disable resource_id`
		3. Log in to nodes with active VSMs and shut down the VSM VMs.
		4. Log in to all three controllers and format the primary_disk and secondary_disk by using the **qemu-img create** *disk-name* **4G** command at */var/spool/cisco/vsm/.*
		5. Enable both the primary and secondary VSMs in pacemaker.
		`#pcs resource enable resource_id`
		6. Recover the missing VSM configuration from backup.
		Note If you cannot log in to one of the VSMs via **virsh console**, use the **peer mac-addresses clear** command on the active VSM accessible through the **virsh console** command.
		Note The VSM configuration might be lost during the VSM split-brain recovery.

Symptom	Possible Causes	Solution

VSM standby is not running.	Pacemaker cannot reinstantiate a primary standby or secondary standby node.	1. Check whether the primary or secondary node is in standby mode. Log in to the active VSM and run the **show redundancy status** command.
		2. Run the **pcs resource cleanup** [*vsm-p*|*vsm-s*] command from one of the controller nodes for the standby VSM.
		3. Check the pacemaker status:
		`[root@overcloud-controller-2`
		`heat-admin]# pcs status | grep vsm`
		`vsm-p  (ocf::heartbeat:VirtualDomain):`
		`Started overcloud-controller-1`
		`vsm-s  (ocf::heartbeat:VirtualDomain):`
		`Started overcloud-controller-2`

VEM configuration is missing.	• Per node specified in the NodeDataLookup configuration is not correctly applied to the node. • The **Puppet apply** command to apply configurations failed to run.	1. Log in to the node with the VEM configuration problem. 2. Run the **dmidecode --s system-uuid** command to retrieve the System UUID. 3. Open the <System-UUID>.json file and confirm whether all configuration parameters expected for the configuration file, *n1kv.conf*, are present. 4. If the parameter values are incorrect, go to the Undercloud and reconfirm the VEM override parameter NodeDataLookup in the configuration file */usr/share/openstack-tripleo-heat-templates/environments/cisco-n1kv-config.yaml*. 5. Perform a heat stack update by redeploying Overcloud to have the latest configurations on the respective nodes.
Virtual Ethernet interfaces on VSM corresponding to the router ports on VEM flap continuously.	The Neutron l3_ha parameter is set to *True* and one or more OpenStack controller nodes (in HA mode) are down or are in inconsistent state.	1. Edit the configuration file, *../etc/neutron/neutron.conf*, on the OpenStack controller node and set the l3_ha parameter to *false* and the allow_automatic_l3agent_failover parameter to *true*. 2. Restart the neutron-server service and the neutron-l3-agent service. 3. Repeat Step 1 and Step 2 on all OpenStack controller nodes in HA. 4. Clean up the router ports by manually deleting and recreating the router ports.

Symptom	Possible Causes	Solution
Inter-VLAN traffic has stopped for VMs.	The Neutron allow_automatic_l3agent_failover parameter is set to *False* and one or more OpenStack controller nodes (in HA mode) are down or are in inconsistent state, causing an automatic migration of l3_agent ports failure.	**1.** Edit the configuration file, *../etc/neutron/neutron.conf*, on the OpenStack controller node and set the allow_automatic_l3agent_failover parameter to *true*. **2.** Restart the neutron-server service and the neutron-l3-agent service. **3.** Repeat Step 1 and Step 2 on all OpenStack controller nodes in HA.
Uplink Ethernet and Virtual Ethernet (VTEP) ports show NoPortProfile state on VSM after deploying using OSP7.	• Port profiles required to bring up ports on VSM are not defined on the VSM. • The port profile names defined in the heat template (/usr/share/openstack-tripleo-heat-templates/environments/cisco-n1kv-config.yaml) do not match the port profile names defined on the VSM.	Create Ethernet and VTEP Virtual Ethernet port profiles with the same name as defined in the heat template (*cisco-n1kv-config.yaml*) on the Undercloud node.
VSM shows the VEM module as offline or a VEM module is missing from VSM **show module** command output after the deployment through OSP7.	The Cisco Nexus 1000V VEM service is not running on the respective VEM module.	Verify the state of the Cisco Nexus 1000V VEM service on the VEM host by using the **service nexus1000v status** command. If the service is not running, restart it by using the **service nexus1000v start** command.
	The Cisco Nexus 1000V VEM cannot communicate with the VSM due to an incorrect networking configuration.	Revisit the network planning. You can leverage the NodeDataLookup parameter to provide node-specific configuration for a single node or class of nodes to enable heterogeneous deployment.
An error is observed on the Openstack controller node when you try to access the policy profiles pushed via the Cisco Nexus 1000V VSM using the **neutron cisco-policy-profile-list** command.	The configuration file, */etc/neutron/Neutron.conf*, is missing a value for the service_plugins parameter.	Update the neutron.conf configuration file with the service_plugins parameter value: `service_plugins =router,cisco_n1kv_profile` After updating the parameter value on all controllers, restart the neutron-server on all controllers in HA mode.

Before Contacting Technical Support

This chapter describes the steps to take before contacting technical support.

Note If you purchased Cisco support through a Cisco reseller, contact the reseller directly. If you purchased support directly from Cisco, contact Cisco Technical Support at this URL: http://www.cisco.com/warp/public/687/Directory/DirTAC.shtm

Cisco Support Communities

For additional information, visit one of the following support communities:

- Cisco Support Community for Server Networking
- Cisco Communities: Nexus 1000V

Gathering Information for Technical Support

At some point, you might need to contact your customer support representative or Cisco TAC for assistance. This section outlines the steps to take before contacting support.

Note Do not reload the module or the switch until you have completed Step 1. Some logs and counters are kept in volatile storage and do not survive a reload.

Step 1 Collect the switch information and configuration before and after the issue has been resolved.

On the VSM, enter the **show tech-support detail > tech-support** command. Use SCP/SFTP/FTP to get the file from the VSM.

On the VEM, enter the following commands in a PowerShell window:

- **set-ExecutionPolicy Unrestricted**
- **cd c:\program files (x86)\Cisco\Nexus1000V\support\**
- **vem-support.ps1**

Add the directory to a zip file to send to technical support.

Step 2 Capture the exact error codes that you see in CLI message logs by entering one of these commands:

- **show logging log** (displays the error messages)
- **show logging last** *number* (displays the last lines of the log)

Step 3 Answer the following questions before contacting technical support:

- On which switch or port is the problem occurring?
- Which Cisco Nexus 1000V software, driver versions, operating systems versions, and storage device firmware are in your fabric?
- What KVM software are you running?
- What is the network topology?
- Were any changes being made to the environment (VLANs, adding modules, upgrades) prior to or at the time of this event?
- Are there other similarly configured devices that could have this problem, but do not?
- Where was this problematic device connected (which switch and interface)?
- When did this problem first occur?
- When did this problem last occur?
- How often does this problem occur?
- How many devices have this problem?
- Were any traces or debug output captured? What troubleshooting steps have you attempted? Which, if any, of the following tools were used:
  - Ethanalyzer, local, or remote SPAN
  - CLI debug commands
  - traceroute, ping

Step 4 Is your problem related to a software upgrade attempt?

- What was the original Cisco Nexus 1000V version?
- What is the new Cisco Nexus 1000V version?

Obtaining a File of Core Memory Information

Cisco customer support engineers often use files from your system for analysis. One file that contains memory information is referred to as a core dump. The file is sent to a TFTP server or to a Flash card in slot0: of the local switch. You should set up your switch to generate this file under the instruction of your customer support representative and send it to a TFTP server so that it can be emailed.

This example shows how to generate a file of core memory information, or a core dump:

```
n1000v# system cores tftp://10.91.51.200/jsmith_cores
n1000v# show system cores
Cores are transferred to tftp://10.91.51.200/jsmith_cores
```

Note The filename (indicated by jsmith_cores) must exist in the TFTP server directory.

Copying Files

You might be required to move files to or from the switch. These files might include log, configuration, or firmware files.

The Cisco Nexus 1000V always acts as a client, so that an ftp/scp/tftp session always originates from the switch and either pushes files to an external system or pulls files from an external system.

```
File Server: 172.22.36.10
File to be copied to the switch: /etc/hosts
```

The **copy** CLI command supports four transfer protocols and 12 different sources for files.

```
n1000v# copy ?
    bootflash: Select source filesystem
    core: Select source filesystem
    debug: Select source filesystem
    ftp: Select source filesystem
    licenses Backup license files
    log: Select source filesystem
    modflash: Select source filesystem
    nvram: Select source filesystem
    running-config Copy running configuration to destination
    scp: Select source filesystem
    sftp: Select source filesystem
    slot0: Select source filesystem
    startup-config Copy startup configuration to destination
    system: Select source filesystem
    tftp: Select source filesystem
    volatile: Select source filesystem
```

Use the following syntax to use secure copy (scp) as the transfer mechanism:

```
"scp:[//[username@]server][/path]"
```

This example shows how to copy /etc/hosts from 172.22.36.10 using the user *user1*, where the destination is hosts.txt:

```
n1000v# copy scp://user1@172.22.36.10/etc/hosts bootflash:hosts.txt
user1@172.22.36.10's password:
hosts 100% |*****************************| 2035 00:00
```

This example shows how to back up the startup configuration to an SFTP server:

```
n1000v# copy startup-config sftp://user1@172.22.36.10/test/startup-configuration.bak1
Connecting to 172.22.36.10...
User1@172.22.36.10's password:
n1000v#
```

Tip Backing up the startup configuration to a server should be done on a daily basis before you make any changes. A short script could be written to be run on the Cisco Nexus 1000V to perform a save and then a backup of the configuration. The script must contain two commands: **copy running-configuration startup-configuration** and **copy startup-configuration tftp**://*server/name*. To run the script, enter the **run-script** *filename* command.

Displaying Licensing Technical Support Information

To display licensing technical support information, use the **show tech-support license** command. For example:

```
n1000v# show tech-support license
`show license host-id`
License hostid: VDH=1234567890123456789
`show license`
`show license usage `
Feature                        Ins Lic   Status Expiry Date Comments
                                   Count
--------------------------------------------------------------------------------
NEXUS_VSG_SERVICES_PKG         No   512  Unused 02 Feb 2014 -
NEXUS1000V_LAN_SERVICES_PKG    No  1024  Unused 02 Feb 2014 -
NEXUS_ASA1000V_SERVICES_PKG    No   512  Unused 02 Feb 2014 -
NEXUS1000V_INTERCLOUD_VM_PKG   No    16  Unused 02 Feb 2014 -
--------------------------------------------------------------------------------
.
.
.
--------------------------------------------------------------------------------
Total bytes: 167360 (163k)
--------------------------------------------------------------------------------
Grand total bytes: 275144 (268k)
```