Cisco Nexus 9000 Series NX-OS Troubleshooting Guide, Release 10.6(x)

First Published: 2025-08-14

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: https://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html. Cisco product warranty information is available at https://www.cisco.com/c/en/us/products/warranty-listing.html. US Federal Communications Commission Notices are found here https://www.cisco.com/c/en/us/products/us-fcc-notice.html.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

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

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

CONTENTS

Preface

This preface includes the following sections:

- Audience, on page xi
- Document Conventions, on page xi
- Related Documentation for Cisco Nexus 9000 Series Switches, on page xii
- Documentation Feedback, on page xii
- Communications, services, and additional information, on page xii

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which you supply the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments that are separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments that are separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string includes the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information that you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

https://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

Communications, services, and additional information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you’re looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions, and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a gateway to the Cisco bug-tracking system, which maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. The BST provides you with detailed defect information about your products and software.

Documentation feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

CHAPTER 1

New and Changed Information

New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 9000 Series NX-OS Troubleshooting Guide, Release 10.6(x)*.

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
Packet Tracer support on Cisco N9336C-SE1 switches	Added packet Tracer support on Cisco N9336C-SE1 switches.	10.6(1)F	Supported Release and Platform for Packet Tracer, on page 105

CHAPTER 2

Overview

Software Image

The Cisco NX-OS software consists of one NXOS software image.

Supported Platforms

Use the Nexus Switch Platform Support Matrix to know from which Cisco NX-OS releases various Cisco Nexus 9000 and 3000 switches support a selected feature.

About the Troubleshooting Process

To troubleshoot your network, follow these general guidelines:

- Maintain a consistent Cisco NX-OS release across all your devices.
- See the Cisco NX-OS release notes for you Cisco NX-OS release for the latest features, limitations, and caveats.
- Enable system message logging.
- Troubleshoot any new configuration changes after implementing the change.
- Gather information that defines the specific symptoms.
- Verify the physical connectivity between your device and end devices.

- Verify the Layer 2 connectivity.
- Verify the end-to-end connectivity and the routing configuration.
- After you have determined that your troubleshooting attempts have not resolved the problem, contact Cisco TAC or your technical support representative.

This section describes the tools that are commonly used to troubleshoot problems within your network.

Note You should have an accurate topology of your network to isolate problem areas. Contact your network architect for this information. Use the following commands to gather general information on your device:

- **show module**
- **show version**
- **show running-config**
- **show logging log**
- **show interfaces brief**
- **show vlan**
- **show spanning-tree**
- **show** {**ip** | **ipv6**} **route**
- **show processes | include ER**
- **show accounting log**

Verifying Ports

Answer the following questions to verify that your ports are connected correctly and are operational:

- Are you using the correct media (copper, optical, fiber type)?
- Is the media broken or damaged?
- Is the port LED green on the module?
- Is the interface operational?

See Troubleshooting Ports for more troubleshooting tips for ports.

Verifying Layer 2 Connectivity

Use the following commands to verify Layer 2 connectivity:

- Use the **show vlan all-ports** command to verify that all the necessary interfaces are in the same VLAN. The status should be active for the VLAN.

- Use the **show port-channel compatibility-parameters** command to verify that all of the ports in a port channel are configured the same for the speed, the duplex, and the trunk mode.
- Use the **show running-config spanning-tree** command to verify that the Spanning Tree Protocol (STP) is configured the same on all devices in the network.
- Use the **show processes | include ER** command to verify that nonessential Layer 2 processes are in the error state.
- Use the **show mac address-table dynamic vlan** command to determine if learning or aging is occurring at each node.

Verifying Layer 3 Connectivity

Answer the following questions to verify Layer 3 connectivity:

- Have you configured a default gateway?
- Have you configured the same dynamic routing protocol parameters throughout your routing domain or configured static routes?
- Are any IP access lists, filters, or route maps blocking route updates?

Use the following commands to verify your routing configuration:

- **show ip arp**
- **show** {**ip** | **ipv6**}
- **show ipv6 neighbor**

Symptoms

This document uses a symptom-based troubleshooting approach that allows you to diagnose and resolve your Cisco NX-OS problems by comparing the symptoms that you observed in your network with the symptoms listed in each chapter.

By comparing the symptoms in this publication to the symptoms that you observe in your own network, you should 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 NX-OS 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.
- Recover from switch upgrade failures.

- Obtain core dumps and other diagnostic data for use by Cisco TAC or your customer support representative.

System Messages

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

System 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].

PORT-3-IF_UNSUPPORTED_TRANSCEIVER: Transceiver for interface [chars] is not supported.

Each system message is followed by an explanation and recommended action. The action may be as simple as "No action is required." It might involve a fix or a recommendation to contact technical support as shown in the following example:

Error Message PORT-3-IF_UNSUPPORTED_TRANSCEIVER: Transceiver for interface [chars] is not supported.

Explanation Transceiver (SFP) is not from an authorized vendor.

Recommended Action Enter the **show interface transceiver** CLI command or similar DCNM command to determine the transceiver being used. Please contact your customer support representative for a list of authorized transceiver vendors.

Syslog Server Implementation

The syslog facility allows the device to send a copy of the message log to a host for more permanent storage. This feature allows you to examine the logs over a long period of time or if the device is not accessible.

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

Syslog uses the facility to determine how to handle a message on the syslog server (the Solaris system in this example) and the message severity. Different message severities are handled differently by the syslog server. They could be logged to different files or e-mailed to a particular user. Specifying a severity level on the syslog server determines that all messages of that level and greater severity (lower number) will be acted upon as you configure the syslog server.

Note You should configure the syslog server so that the Cisco NX-OS messages are logged to a different file from the standard syslog file so that they cannot be confused with other non-Cisco syslog messages. Do not locate the logfile on the / file system. You do not want log messages to fill up the / file system. This example uses the following values:

- syslog client: switch1
- syslog server: 172.22.36.211
- (Solaris) syslog facility: local1
- syslog severity: notifications (level 5, the default)
- File to log Cisco NX-OS messages to: /var/adm/nxos_logs

To configure the syslog feature on Cisco NX-OS, follow these steps:

1. switch# **config terminal**
2. switch(config)# **logging server 192.0.2.1 6 facility local1**

Use the **show logging server** command to verify the syslog configuration.

```
switch1# show logging server
Logging server:                 enabled
{172.22.36.211}
        server severity:        notifications
        server facility:        local1
        server VRF:             management
```

To configure a syslog server, follow these steps:

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

   ```
   local1.notice /var/adm/nxos_logs
   ```

2. Create the log file.

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

3. Restart the syslog process.

   ```
   /etc/init.d/syslog stop
   /etc/init.d/syslog start

   syslog service starting.
   ```

4. Verify that the syslog process has started.

   ```
   ps -ef |grep syslogd
   ```

Test the syslog server by creating an event in Cisco NX-OS. In this case, port e1/2 was shut down and reenabled, and the following was listed on the syslog server. The IP address of the device is listed in brackets.

```
tail -f /var/adm/MDS_logs
Sep 17 11:07:41 [172.22.36.142.2.2] : 2013 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] : 2013 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] : 2013 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

Cisco NX-OS generates many types of system messages on the device and sends them to a syslog server. You can view these messages to determine what events might have led up to the current condition that you are facing.

Use the following commands to access and view logs in Cisco NX-OS:

```
switch# show logging ?
console       Show console logging configuration
info          Show logging configuration
internal      syslog syslog internal information
ip            IP configuration
last          Show last few lines of logfile
level         Show facility logging configuration
logfile       Show contents of logfile
loopback      Show logging loopback configuration
module        Show module logging configuration
monitor       Show monitor logging configuration
nvram         Show NVRAM log
onboard       show logging onboard
server        Show server logging configuration
source-interface  Show logging source-interface configuration
timestamp     Show logging timestamp configuration
```

This example shows the output of the **show logging server** command:

```
switch# show logging server
Logging server:                 enabled
{172.28.254.254}
        server severity:        notifications
        server facility:        local7
        server VRF:             management
```

Troubleshooting Modules

You can directly connect to a module console port to troubleshoot module bootup issues. Use the **attach console module** command to connect to the module console port.

Sometimes a Cisco Nexus End-of-Rack (EoR) switch may fail to boot because of space issue in bootflash. In such a case, verify the free space from the bash shell on the console and remove unnecessary files to get enough free disk space on bootflash. This will ensure smooth boot up of the EoR switch.

Viewing NVRAM Logs

System messages that are priority 0, 1, or 2 are logged into NVRAM on the supervisor module. After a switch reboots, you can display these syslog messages in NVRAM by using the **show logging nvram** command:

```
switch# show logging nvram
2013 Sep 10 15:51:58 switch %$ VDC-1 %$ %SYSMGR-2-NON_VOLATILE_DB_FULL: System n
on-volatile storage usage is unexpectedly high at 99%.
2013 Sep 10 15:52:13 switch %$ VDC-1 %$ %PLATFORM-2-PFM_SYSTEM_RESET: Manual sys
tem restart from Command Line Interface
2013 Sep 10 15:57:49 switch %$ VDC-1 %$ %KERN-2-SYSTEM_MSG: Starting kernel... -
 kernel
2013 Sep 10 15:58:00 switch %$ VDC-1 %$ %CARDCLIENT-2-REG: Sent
2013 Sep 10 15:58:01 switch %$ VDC-1 %$ %USER-1-SYSTEM_MSG: R2D2: P1 SUP NO GMTL
 FOR P1 SUP  - r2d2
2013 Sep 10 15:58:01 switch %$ VDC-1 %$ %USER-1-SYSTEM_MSG: R2D2: P1 SUP NO GMTL
 FOR P1 SUP  - r2d2
2013 Sep 10 15:58:05 switch %$ VDC-1 %$ %USER-1-SYSTEM_MSG: R2D2: P1 SUP: Reset
Tx/Rx during QOS INIT  - r2d2
2013 Sep 10 15:58:16 switch %$ VDC-1 %$ %USER-2-SYSTEM_MSG: can't dlsym ssnmgr_i
s_session_command: please link this binary with ssnmgr.so! - svi
2013 Sep 10 15:58:16 switch %$ VDC-1 %$ %CARDCLIENT-2-SSE: LC_READY sent
2013 Sep 10 15:58:17 switch %$ VDC-1 %$ snmpd: load_mib_module :Error, while loa
ding the mib module /isan/lib/libpmsnmp_common.so (/isan/lib/libpmsnmp_common.so
: undefined symbol: sme_mib_get_if_info)
2013 Sep 10 15:58:17 switch %$ VDC-1 %$ %CARDCLIENT-2-SSE: MOD:6 SUP ONLINE
```

Contacting Customer Support

If you are unable to solve a problem after using the troubleshooting suggestions in this document, contact a customer service representative for assistance and further instructions. Before you call, have the following information ready to help your service provider assist you as quickly as possible:

- Date that you received the device
- Chassis serial number (located on a label on the right side of the rear panel of the chassis)
- Type of software and release number
- Maintenance agreement or warranty information
- Brief description of the problem
- Brief explanation of the steps that you have already taken to isolate and resolve the problem

For more information on steps to take before calling Technical Support, see Steps to Perform Before Calling TAC, on page 137.

CHAPTER 3

Troubleshooting Installations, Upgrades, and Reboots

About Upgrades and Reboots

Upgrades and reboots are ongoing network maintenance activities. You should try to minimize the risk of disrupting the network when performing these operations in production environments and to know how to recover quickly when something does go wrong.

Note This publication uses the term upgrade to refer to both Cisco NX-OS upgrades and downgrades.

Upgrade and Reboot Checklist

Use the following checklist to prepare for an upgrade or reboot:

Checklist	Done
Read the Release Notes for the release to which you are upgrading or downgrading.	
Ensure that an FTP or TFTP server is available to download the software image.	
Copy the new image onto your supervisor modules in bootflash: or slot0:.	
Use the **show install all impact** command to verify that the new image is healthy and the impact that the new load will have on any hardware with regard to compatibility. Check for compatibility.	

Checklist	Done
Copy the startup-config file to a snapshot configuration in NVRAM. This step creates a backup copy of the startup configuration file.	
Save your running configuration to the startup configuration.	
Back up a copy of your configuration to a remote TFTP server.	
Schedule your upgrade during an appropriate maintenance window for your network.	

After you have completed the checklist, you are ready to upgrade or reboot the systems in your network.

Note It is normal for the active supervisor to become the standby supervisor during an upgrade.

Note Up to 100 log messages with a severity level of critical and below (levels 0, 1, and 2) are saved in NVRAM. You can view this log at any time by entering the **show logging nvram** command.

Verifying Software Upgrades

You can use the **show install all status** command to watch the progress of your software upgrade or to view the ongoing **install all** command or the log of the last installed **install all** command from a console, SSH, or Telnet session. This command shows the **install all** output on both the active and standby supervisor module even if you are not connected to the console terminal.

Verifying a Nondisruptive Upgrade

When you initiate a nondisruptive upgrade, Cisco NX-OS notifies all services that an upgrade is about to start and determines whether the upgrade can proceed. If a service cannot allow the upgrade to proceed, the service aborts the upgrade, and you are prompted to enter the **show install all failure-reason** command to determine the reason why the upgrade cannot proceed.

```
Do you want to continue with the installation (y/n)?  [n] y
Install is in progress, please wait.
Notifying services about the upgrade.
>[#                ]   0% -- FAIL. Return code 0x401E0066 (request timed out).
Please issue "show install all failure-reason" to find the cause of the failure.<---prompt
 failure-reason
Install has failed. Return code 0x401E0066 (request timed out).
Please identify the cause of the failure, and try 'install all' again.

switch# show install all failure-reason
Service: "xxx" failed to respond within the given time period.
```

If a failure occurs for any reason (such as a save runtime state failure or a module upgrade failure) after the upgrade is in progress, the device reboots disruptively because the changes cannot be rolled back. In such cases, the upgrade has failed.

If you need further assistance to determine why an upgrade is unsuccessful, you should collect the details from the **show tech-support** [**issu**] command output and the console output from the installation, if available, before you contact your technical support representative.

Troubleshooting Software Upgrades and Downgrades

Software Upgrade Ends with Error

Problem	Possible Cause	Solution
The upgrade ends with an error	The standby supervisor module bootflash: file system does not have sufficient space to accept the updated image.	Use the **delete** command to remove unnecessary files from the file system.
	The **install all** command is entered on the standby supervisor module.	Enter the command on the active supervisor module only.
	A module was inserted while the upgrade was in progress.	Restart the installation.
	The system experienced a power disruption while the upgrade was in progress.	Restart the installation.
	An incorrect software image path was specified.	Specify the entire path for the remote location accurately.
	Another upgrade is already in progress.	Verify the state of the system at every stage and restart the upgrade after 10 seconds. If you restart the upgrade within 10 seconds, the command is rejected. An error message displays, indicating that an upgrade is currently in progress.
	A module failed to upgrade.	Restart the upgrade or use the **install module** command to upgrade the failed module.

Upgrading the Cisco NX-OS Software

You can perform an automated software upgrade on any system from the CLI.

The image filename begins with "nxos" [beginning with Cisco NX-OS Release 7.0(3)I2(1)] or "n9000" (for example, nxos.7.0.3.I2.1.bin or n9000-dk9.7.0.3.I1.1.bin).

Before you begin

Log into the system through the console, Telnet, or SSH port of the active supervisor.

Create a backup of your existing configuration file, if required.

SUMMARY STEPS

1. **install all** [**nxos bootflash:***filename*]
2. **show module**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**install all** [**nxos bootflash:***filename*]	Performs the upgrade. **Note** If the configuration meets all guidelines when the **install all** command is used, all modules (supervisor and switching) are upgraded. **Note** If you enter the **install all** command without specifying a filename, the command performs a compatibility check, notifies you of the modules that will be upgraded, and confirms that you want to continue with the installation. If you choose to proceed, it installs the NXOS software image that is currently running on the switch and upgrades the BIOS of various modules from the running image if required.
Step 2	**show module**	Exits the system console and opens a new terminal session to view the upgraded supervisor module.

Troubleshooting Software System Reboots

Power-On or Switch Reboot Hangs

Problem	Possible Cause	Solution
A power-on or switch reboot hangs for a dual supervisor configuration	The bootflash is corrupted.	See Corrupted Bootflash Recovery, on page 15.
	The BIOS is corrupted.	Replace this module. Contact your customer support representative to return the failed module.
	The nx-os image is corrupted.	Power cycle the switch if required and press **Ctrl-C** when the switch displays the "Loading Boot Loader" message to interrupt the boot process at the >loader prompt.
	Boot parameters are incorrect.	Verify and correct the boot parameters and reboot.

Corrupted Bootflash Recovery

All device configurations reside in the internal bootflash. If you have a corrupted internal bootflash, you could potentially lose your configuration. Be sure to save and back up your configuration files periodically. The regular system boot goes through the following sequence:

1. The basic input/output system (BIOS) loads the loader.
2. The loader loads the nx-os image into RAM and starts the image.
3. The nx-os image reads the startup configuration file.

If the nx-os image on your system is corrupted and you cannot proceed (error state), you can interrupt the system boot sequence and recover the image by entering the BIOS configuration utility described in the following section. Access this utility only when needed to recover a corrupted internal disk.

Caution The BIOS changes explained in this section are required only to recover a corrupted bootflash.

Recovery procedures require the regular sequence to be interrupted. The internal sequence goes through three phases between the time that you turn on the system and the time that the system prompt appears on your terminal—BIOS, boot loader, and nx-os image. The following table describes the steps in the recovery interruption process.

Table 2: Recovery Interruption

Phase	Normal Prompt (appears at the end of each phase)	Recovery Prompt (appears when the system cannot progress to the next phase)	Description
BIOS	loader>	No bootable device	The BIOS begins the power-on self test, memory test, and other operating system applications. While the test is in progress, press **Ctrl-C** to enter the BIOS configuration utility and use the **netboot** option.
Boot loader	Starting nx-os	loader>	The boot loader uncompresses the loaded software to boot an image using its filename as a reference. The image is made available through bootflash. When the memory test is over, press **Esc** to enter the boot loader prompt.

Phase	Normal Prompt (appears at the end of each phase)	Recovery Prompt (appears when the system cannot progress to the next phase)	Description
nx-os image	Uncompressing system	switch(boot)#	When the boot loader phase is over, press **Ctrl-]** (Control key plus right bracket key) to enter the switch(boot)# prompt. Depending on your Telnet client, these keys might be reserved, and you might need to remap the keystroke. See the documentation provided by your Telnet client. If the corruption causes the console to stop at this prompt, copy the nx-os image and reboot the system. The nx-os image then loads the configuration file of the last saved running configuration and returns a switch login prompt.

Recovery from the loader> Prompt

Use the **help** command at the loader> prompt to display a list of commands available at this prompt or to obtain more information about a specific command in that list.

Before you begin

This procedure uses the **init system** command, which reformats the file system of the device. Be sure that you have made a backup of the configuration files before you begin this procedure.

The loader> prompt is different from the regular switch# or switch(boot)# prompt. The CLI command completion feature does not work at the loader> prompt and might result in undesired errors. You must type the command exactly as you want the command to appear.

If you boot over TFTP from the loader> prompt, you must supply the full path to the image on the remote server.

SUMMARY STEPS

1. loader> **set ip** *ip-address*
2. loader> **set gw** *gw-address*
3. loader> **cmdline recoverymode=1**
4. loader> **boot tftp:** *tftp-path*
5. switch(boot)# **init system**
6. switch(boot)# **load-nxos**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	loader> **set ip** *ip-address* **Example:** `loader> set ip 172.21.55.213 255.255.255.224`	Specifies the local IP address and the subnet mask for the system.
Step 2	loader> **set gw** *gw-address* **Example:** `loader> set gw 172.21.55.193`	Specifies the IP address of the default gateway.
Step 3	loader> **cmdline recoverymode=1** **Example:** `loader> cmdline recoverymode=1`	Configures the boot process to stop at the switch(boot)# prompt.

Step 4	loader> **boot tftp:** *tftp-path*
	Example:
	`loader> boot`
	`tftp://172.28.255.18/tftpboot/n9000-dk9.6.1.2.I1.1.bin`
		Boots the nx-os image file from the required server.
		The switch(boot)# prompt indicates that you have a usable nx-os image.

Step 5	switch(boot)# **init system** **Example:** `switch(boot)# init system`	Enters the nx-os system. **Caution** Be sure that you have made a backup of the configuration files before you enter this command.
Step 6	switch(boot)# **load-nxos** **Example:** `switch(boot)# load-nxos`	Completes the upload of the nx-os image file.

Example

This example shows how to configure the local IP address and the subnet mask for the system:

```
loader> set ip 172.21.55.213 255.255.255.224
set ip 172.21.55.213 255.255.255.224
Correct - ip addr is 172.21.55.213, mask is 255.255.255.224
Found Intel 82546GB [2:9.0] at 0xe040, ROM address 0xf980
Probing...[Intel 82546GB]
Management interface
Link UP in 1000/full mode
Ethernet addr: 00:1B:54:C1:28:60
Address: 172.21.55.213
Netmask: 255.255.255.224
Server: 0.0.0.0
Gateway: 172.21.55.193
```

This example shows how to configure the IP address of the default gateway:

```
loader> set gw 172.21.55.193
Correct gateway addr 172.21.55.193
Address: 172.21.55.213
Netmask: 255.255.255.224
Server: 0.0.0.0
Gateway: 172.21.55.193
```

This example shows how to boot the nx-os image from the server:

```
loader> boot tftp://172.28.255.18/tftpboot/n9000-dk9.6.1.2.I1.1.bin
 Address: 172.21.55.213
 Netmask: 255.255.255.224
 Server: 172.28.255.18
 Gateway: 172.21.55.193
  Filesystem type is tftp, using whole disk
 Booting: /tftpboot/n9000-dk9.6.1.2.I1.1.gbin console=ttyS0,9600n8nn quiet loader
 _ver="3.17.0"....
 .............................................................................Im
 age verification OK

 Starting kernel...
 INIT: version 2.85 booting
 Checking all filesystems..r.r.r.. done.
 Setting kernel variables: sysctlnet.ipv4.ip_forward = 0
 net.ipv4.ip_default_ttl = 64
 net.ipv4.ip_no_pmtu_disc = 1
 .
 Setting the System Clock using the Hardware Clock as reference...System Clock set. Local
time: Wed Oct  1
 11:20:11 PST 2013
 WARNING: image sync is going to be disabled after a loader netboot
 Loading system software
 No system image Unexporting directories for NFS kernel daemon...done.
 INIT: Sending processes the KILL signal
 Cisco Nexus Operating System (NX-OS) Software
 TAC support: http://www.cisco.com/tac
 Copyright (c) 2013, Cisco Systems, Inc. All rights reserved.
 The copyrights to certain works contained in this software are
 owned by other third parties and used and distributed under
 license. Certain components of this software are licensed under
 the GNU General Public License (GPL) version 2.0 or the GNU
 Lesser General Public License (LGPL) Version 2.1. A copy of each
 such license is available at
 http://www.opensource.org/licenses/gpl-2.0.php and
 http://www.opensource.org/licenses/lgpl-2.1.php
 switch(boot)#
```

System or Process Restarts

When a recoverable or nonrecoverable error occurs, the system or a process on the system might reset. This table lists possible causes and solutions.

Problem	Possible Cause	Solution
The system or a process on the system resets.	A recoverable error occurred on the system or on a process in the system.	The system has automatically recovered from the problem. See Recovering System Restarts, on page 19.
	A nonrecoverable error occurred on the system.	The system cannot recover automatically from the problem. See Recovering System Restarts, on page 19 to determine the cause.
	A clock module failed.	Verify that a clock module failed. Replace the failed clock module during the next maintenance window.

Recovering System Restarts

Every process restart generates a syslog message and a Call Home event. Even if the event does not affect service, you should identify and resolve the condition immediately because future occurrences could cause a service interruption.

Note After following the steps, determine the cause and resolution for the restart condition by contacting your technical support representative and asking the representative to review your core dump.

Before you begin

The following conditions apply:

- The system automatically copies the core files to a TFTP server every 4 minutes. This time interval is not configurable.
- The copy of a specific core file to a TFTP server can be manually triggered by using the **copy core:**//*module#/pid# tftp://tftp_ip_address/file_name* command.
- If a supervisor failover occurs, the cores might be in the secondary logflash rather than the primary logflash.
- The maximum number of times that a process can be restarted is part of the high-availability (HA) policy for any process. (This parameter is not configurable.) If the process restarts more than the maximum number of times, the older core files are overwritten.
- The maximum number of core files that can be saved for any process is part of the HA policy for any process. (This parameter is not configurable, and it is set to three.)

SUMMARY STEPS

1. switch# **show log | include error**
2. switch# **show processes**
3. switch# **show process log**
4. switch# **show process log pid** *pid*
5. switch# **show system uptime**

6. switch# **show cores**
7. switch# **copy core:** *core path*
8. switch# **show processes log pid** *pid*
9. switch# **system cores tftp:** *tftp-path*

DETAILED STEPS

Procedure

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td><b>Step 1</b></td><td>switch# <b>show log | include error</b>
<b>Example:</b>
<code>switch# show log logfile | include error</code>
<code>Sep 10 23:31:31 dot-6 %</code>
<code>LOG_SYSMGR-3-SERVICE_TERMINATED: Service "sensor"</code>
<code> (PID 704) has finished with error</code>
<code>code SYSMGR_EXITCODE_SY.</code>
<code>switch# show logging logfile | include fail</code>
<code>Jan 27 04:08:42 88 %LOG_DAEMON-3-SYSTEM_MSG: bind()</code>
<code> fd 4, family 2, port 123, ad</code>
<code>dr 0.0.0.0, in_classd=0 flags=1 fails: Address</code>
<code>already in use</code>
<code>Jan 27 04:08:42 88 %LOG_DAEMON-3-SYSTEM_MSG: bind()</code>
<code> fd 4, family 2, port 123, ad</code>
<code>dr 127.0.0.1, in_classd=0 flags=0 fails: Address</code>
<code>already in use</code>
<code>Jan 27 04:08:42 88 %LOG_DAEMON-3-SYSTEM_MSG: bind()</code>
<code> fd 4, family 2, port 123, ad</code>
<code>dr 127.1.1.1, in_classd=0 flags=1 fails: Address</code>
<code>already in use</code>
<code>Jan 27 04:08:42 88 %LOG_DAEMON-3-SYSTEM_MSG: bind()</code>
<code> fd 4, family 2, port 123, ad</code>
<code>dr 172.22.93.88, in_classd=0 flags=1 fails: Address</code>
<code> already in use</code>
<code>Jan 27 23:18:59 88 % LOG_PORT-5-IF_DOWN: Interface</code>
<code> fc1/13 is down (Link failure</code>
<code>or not-connected)</code>
<code>Jan 27 23:18:59 88 % LOG_PORT-5-IF_DOWN: Interface</code>
<code> fc1/14 is down (Link failure</code>
<code>or not-connected)</code>
<code>Jan 28 00:55:12 88 % LOG_PORT-5-IF_DOWN: Interface</code>
<code> fc1/1 is down (Link failure</code>
<code>or not-connected)</code>
<code>Jan 28 00:58:06 88 % LOG_ZONE-2-ZS_MERGE_FAILED:</code>
<code>Zone merge failure, Isolating</code>
<code>port fc1/1 (VSAN 100)</code>
<code>Jan 28 00:58:44 88 % LOG_ZONE-2-ZS_MERGE_FAILED:</code>
<code>Zone merge failure, Isolating</code>
<code>port fc1/1 (VSAN 100)</code>
<code>Jan 28 03:26:38 88 % LOG_ZONE-2-ZS_MERGE_FAILED:</code>
<code>Zone merge failure, Isolating</code>
<code>port fc1/1 (VSAN 100)</code>
<code>Jan 29 19:01:34 88 % LOG_PORT-5-IF_DOWN: Interface</code>
<code> fc1/1 is down (Link failure</code>
<code>or not-connected)</code>
<code>switch#</code></td><td>Displays the syslog file so you can see which process restarted and why it restarted.</td></tr>
</table>

Command or Action | **Purpose**

Step 2

switch# **show processes**

Example:

```
switch# show processes
PID    State  PC        Start_cnt    TTY  Process
-----  -----  --------  -----------  ----
-------------
   1      S  2ab8e33e           1     -   init
   2      S         0           1     -   keventd
   3      S         0           1     -
ksoftirqd_CPU0
   4      S         0           1     -   kswapd
   5      S         0           1     -   bdflush
   6      S         0          1     -   kupdated

  71      S         0          1     -   kjournald

 136      S         0          1     -   kjournald

 140      S         0          1     -   kjournald

 431      S  2abe333e           1     -   httpd
 443      S  2abfd33e           1     -   xinetd
 446      S  2ac1e33e           1     -   sysmgr
 452      S  2abe91a2           1     -   httpd
 453      S  2abe91a2           1     -   httpd
 456      S  2ac73419           1    S0   vsh
 469      S  2abe91a2           1     -   httpd
 470      S  2abe91a2           1     -   httpd
```

Displays the processes that are running and the status of each process.

The following codes are used in the system output for the state (process state):

- 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
ER usually is the state that a process enters if it has been restarted too many times and has been detected as faulty by the system and disabled.

Step 3

switch# **show process log**

Example:

```
switch# show process log
Process PID Normal-exit Stack-trace Core
Log-create-time
------- --- ----------- ----------- ----
---------------
ntp     919      N            N       N   Jan 27
04:08
snsm    972      N            Y       N   Jan 24
20:50
```

Displays the processes that have had abnormal exits and if there is a stack-trace or core dump.

Step 4

switch# **show process log pid** *pid*

Example:

```
switch# show processes log pid 898
Service: idehsd
Description: ide hotswap handler Daemon
Started at Mon Sep 16 14:56:04 2013 (390923 us)
Stopped at Thu Sep 19 14:18:42 2013 (639239 us)
Uptime: 2 days 23 hours 22 minutes 22 seconds
Start type: SRV_OPTION_RESTART_STATELESS (23)
Death reason: SYSMGR_DEATH_REASON_FAILURE_SIGTERM
 (3)
Exit code: signal 15 (no core)
CWD: /var/sysmgr/work
Virtual Memory:
CODE      08048000 - 0804D660
```

Displays detailed information about a specific process that has restarted.

	Command or Action	Purpose

	`    DATA      0804E660 - 0804E824`
	`    BRK       0804E9A0 - 08050000`
	`    STACK     7FFFFD10`
	`Register Set:`
	`EBX 00000003         ECX 0804E994         EDX`
	`00000008`
	`    ESI 00000005         EDI 7FFFFC9C         EBP`
	` 7FFFFCAC`
	`    EAX 00000008         XDS 0000002B         XES`
	` 0000002B`
	`    EAX 00000003 (orig)  EIP 2ABF5EF4         XCS`
	` 00000023`
	`    EFL 00000246         ESP 7FFFFC5C         XSS`
	` 0000002B`
	`Stack: 128 bytes. ESP 7FFFFC5C, TOP 7FFFFD10`
	`0x7FFFFC5C: 0804F990 0804C416 00000003 0804E994`
	`................`
	`0x7FFFFC6C: 00000008 0804BF95 2AC451E0 2AAC24A4`
	`.........Q.*.$.*`
	`0x7FFFFC7C: 7FFFFD14 2AC2C581 0804E6BC 7FFFFCA8`
	`.......*........`
	`0x7FFFFC8C: 7FFFFC94 00000003 00000001 00000003`
	`................`
	`0x7FFFFC9C: 00000001 00000000 00000068 00000000`
	`........h.......`
	`0x7FFFFCAC: 7FFFFCE8 2AB4F819 00000001 7FFFFD14`
	`.......*........`
	`0x7FFFFCBC: 7FFFFD1C 0804C470 00000000 7FFFFCE8`
	`....p...........`
	`0x7FFFFCCC: 2AB4F7E9 2AAC1F00 00000001 08048A2C`
	`...*...*....,...`
	`PID: 898`
	`SAP: 0`
	`UUID: 0`
	`switch#`	

Step 5	switch# **show system uptime**
	Example:
	`switch# show system uptime`
	`Start Time: Fri Sep 13 12:38:39 2013`
	`Up Time:    0 days, 1 hours, 16 minutes, 22 seconds`
		Displays if the restart recently occurred.
		To determine if the restart is repetitive or a one-time occurrence, compare the length of time that the system has been up with the timestamp of each restart.

Step 6	switch# **show cores**
	Example:
	`switch# show cores`
	`Module Instance Process-name PID`
	`Date(Year-Month-Day Time)`
	`------ -------- ------------ ----`
	`-------------------------`
	`28     1        bgp-64551    5179 2013-09-13`
	`23:51:26`
		Displays all cores that are presently available for upload from the active supervisor.

Step 7	switch# **copy core:** *core path* **Example:** `switch# copy core://5/1524 tftp::/1.1.1.1/abcd`	Copies the FSPF core dump to a TFTP server with an IP address.

	Command or Action	Purpose
Step 8	switch# **show processes log pid** *pid* **Example:**	Displays the file named zone_server_log.889 in the log directory,

```
switch# '''show processes log pid 1473'''

=================================================

 Service: ips
 Description: IPS Manager


 Started at Tue Jan  8 17:07:42 2013 (757583 us)
 Stopped at Thu Jan 10 06:16:45 2013 (83451 us)
 Uptime: 1 days 13 hours 9 minutes 9 seconds


 Start type: SRV_OPTION_RESTART_STATELESS (23)
 Death reason: SYSMGR_DEATH_REASON_FAILURE_SIGNAL
 (2)
 Exit code: signal 6 (core dumped)
 CWD: /var/sysmgr/work


 Virtual Memory:


     CODE      08048000 - 080FB060
     DATA      080FC060 - 080FCBA8
     BRK       081795C0 - 081EC000
     STACK     7FFFFCF0
     TOTAL     20952 KB


 Register Set:


     EBX 000005C1         ECX 00000006         EDX
 2AD721E0
     ESI 2AD701A8         EDI 08109308         EBP
 7FFFF2EC
     EAX 00000000         XDS 0000002B         XES
 0000002B
     EAX 00000025 (orig)  EIP 2AC8CC71         XCS
 00000023
     EFL 00000207         ESP 7FFFF2C0         XSS
 0000002B


 Stack: 2608 bytes. ESP 7FFFF2C0, TOP 7FFFFCF0


 0x7FFFF2C0: 2AC8C944 000005C1 00000006 2AC735E2
D..*.........5.*
 0x7FFFF2D0: 2AC8C92C 2AD721E0 2AAB76F0 00000000
,..*.!.*.v.*....
 0x7FFFF2E0: 7FFFF320 2AC8C920 2AC513F8 7FFFF42C
 ... ..*...*,...
 0x7FFFF2F0: 2AC8E0BB 00000006 7FFFF320 00000000
...*.... .......
 0x7FFFF300: 2AC8DFF8 2AD721E0 08109308 2AC65AFC
...*.!.*.....Z.*
```

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td></td><td><pre>
 0x7FFFF310: 00000393 2AC6A49C 2AC621CC 2AC513F8
.......*.!.*...*
 0x7FFFF320: 00000020 00000000 00000000 00000000
 ....
 0x7FFFF330: 00000000 00000000 00000000 00000000
....
 0x7FFFF340: 00000000 00000000 00000000 00000000
....
 0x7FFFF350: 00000000 00000000 00000000 00000000
....
 0x7FFFF360: 00000000 00000000 00000000 00000000
....
 0x7FFFF370: 00000000 00000000 00000000 00000000
....
 0x7FFFF380: 00000000 00000000 00000000 00000000
....
 0x7FFFF390: 00000000 00000000 00000000 00000000
....
 0x7FFFF3A0: 00000002 7FFFF3F4 2AAB752D 2AC5154C
.
 ... output abbreviated ...
 Stack: 128 bytes. ESP 7FFFF830, TOP 7FFFFCD0
</pre></td><td></td></tr>
<tr><td><b>Step 9</b></td><td>switch# <b>system cores tftp:</b> <i>tftp-path</i>
<b>Example:</b>
<code>switch(config)# system cores tftp://10.1.1.1/cores</code></td><td>Configures the system to use TFTP to send the core dump to a TFTP server.
This command causes the system to enable the automatic copy of core files to a TFTP server.</td></tr>
</table>

Unrecoverable System Restarts

An unrecoverable system restart might occur in the following cases:

- A critical process fails and is not restartable.
- A process restarts more times than is allowed by the system configuration.
- A process restarts more frequently than is allowed by the system configuration.

The effect of a process reset is determined by the policy configured for each process. An unrecoverable reset might cause functionality loss, the active supervisor to restart, a supervisor switchover, or the system to restart.

The **show system reset-reason** command displays the following information:

- The last four reset-reason codes for a specific module in a given slot. If a module is absent, the reset-reason codes for that module are not displayed.
- The overall history of when and why expected and unexpected reloads occur.
- The time stamp of when the reset or reload occurred.
- The reason for the reset or reload of a module.
- The service that caused the reset or reload (not always available).
- The software version that was running at the time of the reset or reload.

```
switch# show system reset-reason module 27
----- reset reason for Supervisor-module 27 (from Supervisor in slot 27) ---
1) At 281000 usecs after Wed Jun 26 20:16:34 2013
    Reason: Reset Requested by CLI command reload
    Service:
    Version: 6.1(2)I1(1)
2) At 791071 usecs after Wed Jun 26 20:04:50 2013
    Reason: Reset Requested by CLI command reload
    Service:
    Version: 6.1(2)I1(1)
3) At 70980 usecs after Wed Jun 26 19:55:52 2013
    Reason: Reset Requested by CLI command reload
    Service:
    Version: 6.1(2)I1(1)
4) At 891463 usecs after Wed Jun 26 23:44:48 2013
    Reason: Reset Requested by CLI command reload
    Service:
    Version: 6.1(2)I1(1)
```

Standby Supervisor Fails to Boot

The standby supervisor does not boot after an upgrade. You may see the following system message:

```
SYSMGR-2-STANDBY_BOOT_FAILED
```

This message is printed if the standby supervisor does not complete its boot procedure (does not reach the login prompt on the local console) 3 to 6 minutes after the loader has been loaded by the BIOS. This message is usually caused by boot variables not properly set for the standby supervisor. This message can also be caused by a user intentionally interrupting the boot procedure at the loader prompt (by pressing ESC).

Connect to the local console of the standby supervisor. If the supervisor is at the loader prompt, try to use the **boot** command to continue the boot procedure. Otherwise, enter the **reload** command for the standby supervisor from a vsh session on the active supervisor, specifying the **force-dnld** option. Once the standby is online, fix the problem by setting the boot variables appropriately.

Symptom	Possible Cause	Solution
Standby supervisor does not boot.	Active supervisor nx-os image booted from TFTP.	Reload the active supervisor from bootflash:.

Recovering the Administrator Password

See Recovering the Administrator Password topic in Password Recovery Procedure for Cisco NX-OS guide to know how to recover administrator password.

- From the CLI with a username that has network-admin privileges
- By power cycling the device
- By reloading the device

Using the CLI with Network-Admin Privileges to Recover the Administrator Password

SUMMARY STEPS

1. switch# **show user-account**

2. switch# **config terminal**
3. switch(config)# **username admin password** *new-password*
4. switch(config)# **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	switch# **show user-account**
	Example:
	`switch# show user-account`
	`user:admin`
	`        this user account has no expiry date`
	`        roles:network-admin`
	`user:dbgusr`
	`        this user account has no expiry date`
	`        roles:network-admin network-operator`
		Shows that your username has network-admin privileges.

Step 2	switch# **config terminal**
	Example:
	`switch# config terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 3	switch(config)# **username admin password** *new-password* **Example:** `switch(config)# username admin password egBdf`	Assigns a new network administrator password if your username has network-admin privileges. **Note** The *new-password* does not allow the $ character.
Step 4	switch(config)# **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Power Cycling the Device to Recover the Administrator Password

If you cannot start a session on the device that has network-admin privileges, you can recover the network administrator password by power cycling the device.

Caution The password recovery procedure disrupts all traffic on the device. All connections to the device will be lost for 2 to 3 minutes.

Note You cannot recover the administrator password from a Telnet or Secure Shell (SSH) session to the management interface. You must have access to the local console connection.

Note Password recovery updates the new administrator password only in the local user database and not on the remote AAA servers. The new password works only if local authentication is enabled; it does not work for remote authentication. When a password is recovered, local authentication is enabled for logins through a console so that the admin user can log in with a new password from a console.

Note If you need to recover the password because the username was not specified in the configuration file when you performed a **copy** *configuration-file* **startup-config** followed by the **fast-reload** or **reload** command, you will need to perform a **write erase** in Step 12 below.

Before you begin

On a device with two supervisor modules, you must perform the password recovery procedure on the supervisor module that will become the active module after you complete the recovery procedure. To ensure that the other supervisor module does not become active, perform one of the following tasks:

- Remove the other supervisor module from the chassis.
- Change the console prompt of the other supervisor module to one of the following two prompts until the recovery procedure completes:
  - loader >
  - switch(boot)#

Procedure

	Command or Action	Purpose
Step 1	Establish a terminal session on the console port of the active supervisor module.	— **Note** If you are using a non-U.S. keymap, the key sequence that you need to press to generate the break sequence might not work. In this case, we recommend that you set your terminal to a U.S. keymap. You can enter **Ctrl-C** instead of **Ctrl-]** (right square bracket) due to keyboard mapping.
Step 2	If you use SSH or a terminal emulator to access the console port, go to Step 6.	—

	Command or Action	Purpose
Step 3	If you use Telnet to access the console port, press **Ctrl-]** (right square bracket) to verify that it does not conflict with the Telnet escape sequence. **Example:** `switch login: Ctrl-]`	— **Note** If the Cisco NX-OS login prompt remains and the Telnet prompt does not appear, go to Step 6.

Step 4	If the Telnet prompt appears, change the Telnet escape sequence to a character sequence other than Ctrl-] (right square bracket).
	Example:
	`telnet> set escape ^\`
	`Escape Character is 'CTRL+\'`
		The example shows how to set Ctrl-\ as the escape key sequence in Microsoft Telnet.
		Note
		If the Cisco NX-OS login prompt remains and the Telnet prompt does not appear, go to Step 6.

Step 5	Press **Enter** one or more times to return to the Cisco NX-OS login prompt.
	Example:
	`telnet> <Enter>`
	`switch login:`
		—

Step 6	Power cycle the device.	—

Step 7	Press **Ctrl-C** to access the loader> prompt.
	Example:
	`Ctrl-C`
	`loader>`
		—

Step 8	loader> **cmdline recoverymode=1** **Example:** `loader> cmdline recoverymode=1`	Enters recovery mode.

Step 9	loader> **boot n9000-dk9.x.x.x.bin**
	Example:
	`loader> boot n9000-dk9.x.x.x.bin`
	`Booting iash`
	`Trying diskboot`
	` Filesystem type is ext2fs, partition type 0x83`
	`Image valid`
	`MD5Sum mismatch`
	
	`INIT: Loading IGB driver ... Signature`
	`Envelope.(36)Invalid Tag in Signature Envelope`
	`Installing SSE module ... done`
	`Creating the sse device node ... done`
	`Installing CCTRL driver for card_type 3 ...`
		Restarts the device with the nx-os image to reach the switch(boot)# prompt.

	Command or Action	Purpose

	`Checking all filesystems.......`
	`Installing SPROM driver ...`
	`Installing default sprom values ...`
	` done.Configuring network ...`
	`Installing psdev ...`
	`Installing veobc ...`
	`Installing OBFL driver ...`
	`Starting portmap daemon...`
	`creating NFS state directory: done`
	`starting 8 nfsd kernel threads: done`
	`starting mountd: done`
	`starting statd: done`
	`Loading system software`
	`No system image is specified`
	`INIT: Sending processes the TERM signal`
	`INIT: Sending processes the KILL signal`
	`Bad terminal type: "linux". Will assume vt100.`
	`Cisco Nexus Operating System (NX-OS) Software`
	`TAC support: http://www.cisco.com/tac`
	`Copyright (c) 2002-2013, Cisco Systems, Inc. All`
	`rights reserved.`
	`The copyrights to certain works contained in this`
	`software are owned by other third parties and used`
	`and distributed under license. Certain components`
	`of this software are licensed under the GNU`
	`General`
	`Public License (GPL) version 2.0 or the GNU`
	`Lesser General Public License (LGPL) Version 2.1.`
	`A copy of each such license is available at`
	`http://www.opensource.org/licenses/gpl-2.0.php`
	`and`
	`http://www.opensource.org/licenses/lgpl-2.1.php`
	`switch(boot)#`	

Step 10	Press **Enter** one or more times to return to the Cisco NX-OS login prompt.
	Example:
	`telnet> <Enter>`
	`switch login:`
		—

Step 11	switch(boot)# **config terminal**
	Example:
	`switch(boot)# config terminal`
	`Enter configuration commands, one per line.  End`
	`with CNTL/Z.`
	`switch(boot)(config)#`
		Enters boot configuration mode.

Step 12	switch(boot)(config)# **admin-password** *new-password*
	Example:
	`switch(boot)(config)# admin-password egBdf`
	`WARNING! Remote Authentication for login through`
	` console has been disabled`
		Resets the network administrator password.
		Note
		If you are performing this password recovery procedure because the username was not specified in the configuration file when you performed a **copy** *configuration-file* **startup-config** followed by the **fast-reload** or **reload** command, skip this step, enter the

	Command or Action	Purpose
		write erase command instead, and then go to the next step. **Important** If your switch is running Cisco NX-OS Release 7.0(3)I2(2), skip Steps 12 through 14, perform a write erase, and reload the device. Make sure that the configurations are backed up before attempting the password recovery. This workaround pertains only to Cisco NX-OS Release 7.0(3)I2(2).

Step 13	switch(boot)(config)# **exit**
	Example:
	`switch(boot)(config)# exit`
	`switch(boot)#`
		Exits boot configuration mode.

Step 14	switch(boot)# **load-nxos** **Example:** `switch(boot)# load-nxos`	Loads the nx-os image. You must enter the **load-nxos** command exactly as shown. Do not enter the image filename with this command.

Step 15	Log into the device using the new administrator password.
	Example:
	`switch login: admin`
	`Password: egBdf`
		The running configuration indicates that local authentication is enabled for logins through a console. You should not change the running configuration in order for the new password to work for future logins. You can enable remote authentication after you reset and remember the administrator password that is configured on the AAA servers.
		`switch# show running-config aaa`
		`!Command: show running-config aaa`
		`!Time: Fri Jun 7 02:39:23 2013`
		`version 6.1(2)I1(1)`
		`logging level aaa 5`
		`aaa authentication login ascii-authentication`

Step 16	switch# **config terminal**
	Example:
	`switch# config terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 17	switch(config)# **username admin password** *new-password* **Example:** `switch(config)# username admin password egBdf`	Resets the new password to ensure that it is also the Simple Network Management Protocol (SNMP) password.

	Command or Action	Purpose

Step 18	switch(config)# **exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 19	Insert the previously removed standby supervisor module into the chassis, if necessary.	—
Step 20	Boot the nx-os image on the standby supervisor module, if necessary.	—
Step 21	switch(config)# **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Reloading the Device to Recover the Administrator Password

You can reset the network administrator password by reloading the device.

Caution This procedure disrupts all traffic on the device. All connections to the device will be lost for 2 to 3 minutes.

Note You cannot recover the administrator password from a Telnet or Secure Shell (SSH) session to the management interface. You must have access to the local console connection.

Note Password recovery updates the new administrator password only in the local user database and not on the remote AAA servers. The new password works only if local authentication is enabled; it does not work for remote authentication. When a password is recovered, local authentication is enabled for logins through a console so that the admin user can log in with a new password from a console.

SUMMARY STEPS

1. Establish a terminal session on the console port of the active supervisor module.
2. switch# **reload**
3. loader> **cmdline recoverymode=1**
4. loader> **boot n9000-dk9.x.x.x.bin**
5. Reset the network administrator password by following Steps 6 through 20 in Power Cycling the Device to Recover the Administrator Password, on page 26.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	Establish a terminal session on the console port of the active supervisor module.	—

Step 2	switch# **reload**
	Example:
	`switch# reload`
	`This command will reboot the system. (y/n)?  [n]`
	`Y`
	`2013 Jun  7 13:09:56 switch %$ VDC-1 %$`
	`%PLATFORM-2-PFM_SYSTEM_RESET:`
	`Manual system restart from Command Line Interface`
	` writing reset reason 9,`
	`..`
	`..`
	`              GNU GRUB  version 0.97`
	`Autobooting bootflash:/n9000-dk9.x.x.x.bin`
	`bootflash:/n...`
	` Filesystem type is ext2fs, partition type 0x83`
	`Booting nx-os image:`
	`bootflash:/n9000-dk9.x.x.x.bin....(---->`
	`Press Ctrl + C)`
	`....Aborting Image Boot`
	`              GNU GRUB  version 0.97`
	`                Loader Version 3.22.0`
	`loader>`
		Reloads the device to reach the loader prompt. You need to press **`Ctrl-C`** when the following appears:
		`Booting nx-os image:`
		`bootflash:/n9000-dk9.x.x.x.bin....`

Step 3	loader> **cmdline recoverymode=1** **Example:** `loader> cmdline recoverymode=1`	Configure the boot process to stop at the switch(boot)# prompt.

Step 4	loader> **boot n9000-dk9.x.x.x.bin**
	Example:
	`loader> boot n9000-dk9.x.x.x.bin`
	`Filesystem type is ext2fs, partition type 0x83`
	`Booting nx-os image: n9000-dk9.6.1.2.I1.1.gbin....`
	`................................................`
	`.....................Image verification OK`
	`..`
	`..`
	`Lesser General Public License (LGPL) Version 2.1.`
	`A copy of each such license is available at`
	`http://www.opensource.org/licenses/gpl-2.0.php and`
	`http://www.opensource.org/licenses/lgpl-2.1.php`
	`switch(boot)#`
		Restarts the device with only the nx-os image to reach the switch boot prompt.

Step 5	Reset the network administrator password by following Steps 6 through 20 in Power Cycling the Device to Recover the Administrator Password, on page 26.	—

Changing the Administrator Password

You must be logged in as admin to change the network administrator password.

Guidelines and Limitations for Changing the Administrator Password

Follow these guidelines and limitations to change an administrator password:

- You must be an admin to enable or disable the CLI command, no service password-recovery.
- You must be logged in as admin to change the admin password.
- You cannot change the admin password from a boot prompt if the CLI was disabled by the admin on a previous boot.

Note If you are not logged in as admin, you see an error.

Granting the Change Admin Password to Admin User Only

SUMMARY STEPS

1. switch# **show user-account**
2. switch# **configure terminal**
3. switch(config)# **no service password-recovery**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	switch# **show user-account**
	Example:
	`switch# show user-account`
	`user:admin`
	`        this user account has no expiry date`
	`        roles:network-admin`
	`user:dbgusr`
	`        this user account has no expiry date`
	`        roles:network-admin network-operator`
		Shows that your username has network-admin privileges.

Step 2	switch# **configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 3	switch(config)# **no service password-recovery**
	Example:
	`switch(config)# no service password-recovery`
	`WARNING: executing this command will disable the`
	`password recovery mechanism. Do not execute this`
	`command without another plan for password recovery.`
	` Are you sure you want to continue? (y/n) : [y] y`
		Enables/disables password recovery.
		Note
		To allow another user to change the Admin password, run **service password-recovery** when logged in as admin with network admin privileges.

CHAPTER 4

Troubleshooting Licensing Issues

About Troubleshooting Licensing Issues

Cisco NX-OS requires licenses for select features. The licenses enable those features on your system. You must purchase a license for each system on which you want to enable the licensed features.

Chassis Serial Numbers

Licenses are created using the serial number of the chassis where the license file is to be installed. Once you order a license based on a chassis serial number, you cannot use this license on any other system.

Swapping out a Chassis

If you swap out a chassis which included licenses, you must contact TAC to generate a new license. The old license was based on the chassis serial number and will not work with the new chassis.

Guidelines and Limitations for Licensing

Follow these guidelines when dealing with licenses for Cisco NX-OS:

- Carefully determine the license(s) that you require based on the features that require a license.
- Order your license accurately, as follows:
  - Enter the Product Authorization Key that appears in the Proof of Purchase document that comes with your system.
  - Enter the correct chassis serial number when ordering the license. The serial number must be for the same chassis on which you plan to install the license. Use the **show license host-id** command to obtain your chassis serial number.
  - Enter serial numbers accurately. Do not use the letter "O" instead of a zero in the serial number.

  - Order the license that is specific to your chassis.

- Back up the license file to a remote, secure place. Archiving your license files ensures that you will not lose the licenses in the case of a failure on your system.
- Install the correct licenses on each system, using the licenses that were ordered using that system's serial number. Licenses are serial-number specific and platform specific.
- Use the **show license usage** command to verify the license installation.
- Never modify a license file or attempt to use it on a system for which it was not ordered. If you return a chassis, contact your customer support representative to order a replacement license for the new chassis.

Initial Troubleshooting Checklist for Licensing

Begin troubleshooting license issues by checking the following issues first:

Checklist	Done
Verify the chassis serial number for all licenses ordered.	
Verify the platform or module type for all licenses ordered.	
Verify that the Product Authorization Key that you used to order the licenses comes from the same chassis from which you retrieved the chassis serial number.	
Verify that you have installed all licenses on all systems that require the licenses for the features you enable.	

Displaying License Information Using the CLI

SUMMARY STEPS

1. **show license** [**host-id** \| **usage** [*package*]]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**show license** [**host-id** | **usage** [*package*]] **Example:** `switch# show license usage LAN_ENTERPRISE_SERVICES_PKG`	Displays license information configured on this system. Use the **host-id** keyword to display the host ID for the license. Use the **usage** keyword to display a list of all licensed features or a list of features in a specified package.

Example

This example displays all installed license key files and contents:

```
switch# show license
entp.lic:
SERVER this_host ANY
VENDOR cisco
INCREMENT LAN_ENTERPRISE_SERVICES_PKG cisco 1.0 permanent uncounted \
        VENDOR_STRING=<LIC_SOURCE>MDS_SWIFT</LIC_SOURCE><SKU>N95-LAN1K9=</SKU> \
        HOSTID=VDH=TBC10412106 \ >
        NOTICE="<LicFileID>20071025133322456</LicFileID>LicLineID>1/LicLineID>
 \
```

This example displays information about current license usage:

```
switch# show license usage
Feature                      Ins  Lic  Status   Expiry Date Comments        Count
------------------------------------------------------------------------------------------

LAN_ENTERPRISE_SERVICES_PKG   No    -   In use
------------------------------------------------------------------------------------------
```

This example displays a list of features in a specified package:

```
switch# show license usage LAN_ENTERPRISE_SERVICES_PKG
Application
-----------
bgp
pim
msdp
ospf
ospfv3
-----------
```

This example displays the host ID for the license:

```
switch# show license host-id
License hostid: VDH=FOX0646S017
```

Note Use the entire ID that appears after the colon (:) . The VHD is the Vendor Host ID.

Licensing Installation Issues

Serial Number Issues

Make sure that you use the correct chassis serial number when ordering your license. Use the **show license host-id** command to obtain the correct chassis serial number for your system using the CLI.

If you use a license meant for another chassis, you might see the following system message:

Error Message: `LICMGR-3-LOG_LIC_INVALID_HOSTID: Invalid license hostid VDH=[chars] for feature [chars].`

Explanation: `The feature has a license with an invalid license Host ID. This can happen if a supervisor module with licensed features for one system is installed on another system.`

Recommended Action: `Reinstall the correct license for the chassis where the supervisor module is installed.`

Note When entering the chassis serial number during the license ordering process, do not use the letter "O" instead of any zeros in the serial number.

RMA Chassis Errors or License Transfers Between Systems

A license is specific to the system for which it is issued and is not valid on any other system. If you need to transfer a license from one system to another, contact your technical support representative.

License Listed as Missing

After a license is installed and operating properly, it might show up as missing if you modify your system hardware or encounter a bootflash: issue.

Symptom	Possible Causes	Solutions
A license is listed as missing.	The supervisor module was replaced after the license was installed.	See Corrupted Bootflash Recovery, on page 15 to recover from the corrupted bootflash:. Reinstall the license.
	The supervisor bootflash: is corrupted.	

CHAPTER 5

Troubleshooting Ports

About Troubleshooting Ports

Before a device can relay frames from one data link to another, the characteristics of the interfaces through which the frames are received and sent must be defined. The configured interfaces can be Ethernet interfaces, VLAN interfaces (SVIs), or the management interface (mgmt0).

Each interface has an associated administrative configuration and operational status as follows:

- The administrative configuration does not change unless you modify it. This configuration has various attributes that you can configure in administrative mode.
- The operational status represents the current status of a specified attribute such as the interface speed. This status cannot be changed and is read-only. Some values may not be valid when the interface is down (such as the operation speed).

For a complete description of port modes, administrative states, and operational states, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.

Guidelines and Limitations for Troubleshooting Ports

Follow these guidelines when you configure a port interface:

- Before you begin configuring a device, make sure that the modules in the chassis are functioning as designed. Use the **show module** command to verify that a module is OK or active before continuing the configuration.
- When configuring dedicated ports in a port group, follow these port mode guidelines:
  - You can configure only the one port in each four-port group in dedicated mode. The other three ports are not usable and remain shut down.

  - If any of the other three ports are enabled, you cannot configure the remaining port in dedicated mode. The other three ports continue to remain enabled.

- There are no licensing requirements for port configuration in Cisco NX-OS.

Initial Port Troubleshooting Checklist

Begin troubleshooting the port configuration by checking the following issues:

Checklist	Done
Check the physical media to ensure that there are no damaged parts.	
Verify that the SFP (small form-factor pluggable) devices in use are those authorized by Cisco and that they are not faulty.	
Verify that you have enabled the port by using the **no shutdown** command.	
Use the **show interface** command to verify the state of the interface. See the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide* for reasons why a port might be in a down operational state.	
Verify that you have configured a port as dedicated and make sure that you have not connected to the other three ports in the port group.	

Viewing Port Information

You can use the **show interface counters** command to view port counters. Typically, you only observe counters while actively troubleshooting, in which case you should first clear the counters to create a baseline. The values, even if they are high for certain counters, can be meaningless for a port that has been active for an extended period. Clearing the counters provides a better idea of the link behavior as you begin to troubleshoot.

Use one of the following commands to clear all port counters or the counters for specified interfaces:

- **clear counters interface all**
- **clear counters interface** *range*

The counters can identify synchronization problems by displaying a significant disparity between received and transmitted frames.

Use the following commands to gather more information about ports:

- **show interface status**
- **show interface capabilities**
- **show udld**
- **show tech-support udld**

Troubleshooting Port Statistics from the CLI

To display complete information for an interface, use the **show interface** command. In addition to the state of the port, this command displays the following:

- Speed
- Trunk VLAN status
- Number of frames sent and received
- Transmission errors, including discards, errors, and invalid frames

```
switch# show interface ethernet 2/45
Ethernet2/45 is down (Administratively down)
  Hardware is 10/100/1000 Ethernet, address is 0019.076c.4dd8 (bia 0019.076c.4dd8)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  auto-duplex, auto-speed
  Beacon is turned off
  Auto-Negotiation is turned on
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned on
  Last clearing of "show interface" counters never
  1 minute input rate 0 bytes/sec, 0 packets/sec
  1 minute output rate 0 bytes/sec, 0 packets/sec
  L3 Switched:
    input: 0 pkts, 0 bytes - output: 0 pkts, 0 bytes
  Rx
    0 input packets 0 unicast packets 0 multicast packets
    0 broadcast packets 0 jumbo packets 0 storm suppression packets
    0 bytes
  Tx
    0 output packets 0 multicast packets
    0 broadcast packets 0 jumbo packets
    0 bytes
    0 input error 0 short frame 0 watchdog
    0 no buffer 0 runt 0 CRC 0 ecc
    0 overrun  0 underrun 0 ignored 0 bad etype drop
    0 bad proto drop 0 if down drop 0 input with dribble
    0 output error 0 collision 0 deferred
    0 late collision 0 lost carrier 0 no carrier
    0 babble
    0 Rx pause 0 Tx pause 0 reset
Receive data field Size is 2112
```

Troubleshooting Port-Interface Issues

The Interface Configuration Has Disappeared

You may have a problem where your interface configuration disappears.

Symptoms	Possible Cause	Solution
The interface configuration has disappeared.	The interface mode has changed to or from the switchport mode.	Cisco NX-OS removes the interface configuration when you switch between Layer 2 and Layer 3 port mode. You must reconfigure the interface.

You Cannot Enable an Interface

You might have a problem when enabling an interface.

Problem	Possible Cause	Solution
You cannot enable an interface.	The interface is part of a dedicated port group.	You cannot enable the other three ports in a port group if one port is dedicated. Use the **show running-config interface** CLI command to verify the rate mode setting.
	The interface configuration is incompatible with a remote port.	Use the **show interface capabilities** command on both ports to determine if both ports have the same capabilities. Modify the configuration as needed to make the ports compatible.
	The Layer 2 port is not associated with an access VLAN, or the VLAN is suspended.	Use the **show interface brief** command to see if the interface is configured in a VLAN. Use the **show vlan brief** command to determine the status of the VLAN. Use the **state active** command in VLAN configuration mode to configure the VLAN as active.
	An incorrect SFP is connected to the port.	Use the **show interface brief** command to see if you are using an incorrect transceiver. Replace with a Cisco-supported SFP.

You Cannot Configure a Dedicated Port

You may have a problem when trying to configure a port as dedicated.

Problem	Possible Cause	Solution
You cannot configure a dedicated port.	The other three ports in the port group are not shut down.	Use the **shutdown** command in interface configuration mode to disable the other three ports in the port group.
	The port is not the first port in the port group.	You can set only the first port in a port group to the dedicated mode.

A Port Remains in a Link Failure or Not Connected State

You may have a problem with ports or links becoming operational.

Problem	Possible Cause	Solution
A port remains in a link-failure state.	The port connection is bad.	Use the **show port internal info** command to verify the port status is in link-failure. Verify the type of media in use. Is it optical, single-mode (SM), or multimode (MM)? Use the **shutdown** command followed by the **no shutdown** command to disable and enable the port. If this problem persists, try moving the connection to a different port on the same or another module.
	There is no signal because of a transit fault in the small form-factor pluggable (SFP), or the SFP may be faulty.	When this problem occurs, the port stays in a transit port state and you see no signal. There is no synchronization at the MAC level. The problem might be related to the port speed setting or autonegotiation. Verify that the SFP on the interface is seated properly. If reseating the SFP does not resolve the issue, replace the SFP or try another port on the switch.
	The link is stuck in the initialization state, or the link is in a point-to-point state.	Use the **show logging** command to check for a "Link Failure, Not Connected system" message. Use the **shutdown** command followed by the **no shutdown** command to disable and enable the port. If this problem persists, try moving the connection to a different port on the same or another module.

An Unexpected Link Flapping Occurs

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.

When you are troubleshooting an unexpected link flapping, you should know the following information:

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

Problem	Possible Cause	Solution
An unexpected link flapping occurs.	The bit rate exceeds the threshold and puts the port into the errDisabled state.	Use the **shutdown** command followed by the **no shutdown** command to return the port to the normal state.
	A problem in the system triggers the link flap action by the end device. Some of the causes are as follows: • A packet drop in the device occurs because of either a hardware failure or an intermittent hardware error such as an X-bar sync loss. • A packet drop results from a software error. • A control frame is erroneously sent to the device.	Determine the link flap reason as indicated by the MAC driver. Use the debug facilities on the end device to troubleshoot the problem. An external device might choose to reinitialize the link when it encounters the error. In such cases, the method of reinitializing the link varies by device.

A Port Is in the ErrDisabled State

The ErrDisabled state indicates that the switch detected a problem with the port and disabled the port. This state could be caused by a flapping port which could indicate a problem with the media.

Problem	Possible Cause	Solution
A port is in the ErrDisabled state.	The port is flapping.	See Verifying the ErrDisable State Using the CLI, on page 44 to verify the SFP, cable, and connections.

Verifying the ErrDisable State Using the CLI

SUMMARY STEPS

1. switch# **show interface** *interface slot*/*port*
2. switch# **show system internal etphm event-history interface** *interface slot*/*port*
3. switch# **show logging logfile**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **show interface** *interface slot*/*port* **Example:**	Verifies that the device detected a problem and disabled the port. **Note**

	Command or Action	Purpose

	`switch# show interface ethernet 1/14`
	`e1/7 is down (errDisabled)`
		After verifying the port is disabled, check cables, SFPs, and optics.

Step 2	switch# **show system internal etphm event-history interface** *interface slot*/*port*
	Example:
	`switch# show system internal ethpm event-history`
	`interface`
	`ethernet 1/7`
		Displays information about the internal state transitions of the port.

Step 3	switch# **show logging logfile** **Example:** `switch# show logging logfile`	Displays the switch log file and view a list of port state changes.

Example

This example shows how view information about the internal state transitions of the port. The port ethernet 1/7 entered the ErrDisabled state because of a capability mismatch, or "CAP MISMATCH":

```
switch# show system internal ethpm event-history interface
ethernet 1/7
>>>>FSM: <e1/7> has 86 logged transitions<<<<<
1) FSM:<e1/7> Transition at 647054 usecs after Tue Jan  1 22:44..
    Previous state: [ETH_PORT_FSM_ST_NOT_INIT]
    Triggered event: [ETH_PORT_FSM_EV_MODULE_INIT_DONE]
    Next state: [ETH_PORT_FSM_ST_IF_INIT_EVAL]
2) FSM:<e1/7> Transition at 647114 usecs after Tue Jan  1 22:43..
    Previous state: [ETH_PORT_FSM_ST_INIT_EVAL]
    Triggered event: [ETH_PORT_FSM_EV_IE_ERR_DISABLED_CAP_MISMATCH]
    Next state: [ETH_PORT_FSM_ST_IF_DOWN_STATE]
```

This example shows how to display the switch log file and view a list of port state changes. An error was recorded when someone attempted to add port e1/7 to port channel 7. The port was not configured identically to port channel 7, so the attempt failed:

```
switch# show logging logfile
 . . .
Jan  4 06:54:04 switch %PORT_CHANNEL-5-CREATED: port-channel 7 created
Jan  4 06:54:24 switch %PORT-5-IF_DOWN_PORT_CHANNEL_MEMBERS_DOWN: Interface
port-channel 7 is down (No operational members)
Jan  4 06:54:40 switch %PORT_CHANNEL-5-PORT_ADDED: e1/8 added to port-channel 7
Jan  4 06:54:56 switch %PORT-5-IF_DOWN_ADMIN_DOWN: Interface e1/7 is down
(Admnistratively down)
Jan  4 06:54:59 switch %PORT_CHANNEL-3-COMPAT_CHECK_FAILURE:
speed is not compatible
Jan 4 06:55:56 switch%PORT_CHANNEL-5-PORT_ADDED: e1/7 added to port-channel 7
```

CHAPTER 6

Troubleshooting vPCs

About Troubleshooting vPCs

A vPC allows links that are physically connected to two different devices to appear as a single port channel by a third device.

Initial Troubleshooting vPCs Checklist

Begin troubleshooting vPC issues by checking the following issues first:

Checklist	Done
Is the vPC keepalive link mapped to a separate VRF? If not, it will be mapped to the management VRF by default. In this case, do you have a management switch connected to the management ports on both vPC peer devices?	
Verify that both the source and destination IP addresses used for the peer-keepalive messages are reachable from the VRF associated with the vPC peer-keepalive link.	
Verify that the peer-keepalive link is up. Otherwise, the vPC peer link will not come up.	
Verify that the vPC peer link is configured as a Layer 2 port channel trunk that allows only vPC VLANs.	
Verify that the vPC number that you assigned to the port channel that connects to the downstream device from the vPC peer device is identical on both vPC peer devices.	
If you manually configured the system priority, verify that you assigned the same priority value on both vPC peer devices.	

Checklist	Done
Check the **show vpc consistency-parameters** command to verify that both vPC peer devices have identical type-1 parameters.	
Verify that the primary vPC is the primary STP root and the secondary vPC is the secondary STP root.	

Verifying vPCs Using the CLI

To verify vPCs using the CLI, perform one of these tasks:

Command	Purpose
show running-config vpc	Verifies the vPC configuration.
show vpc	Checks the status of the vPCs.
show vpc peer-keepalive	Checks the status of the vPC peer-keepalive link.
show vpc consistency-parameters	Verifies that the vPC peers have the identical type-1 parameters.
show tech-support vpc	Displays detailed technical support information for vPCs.
show port-channel summary	Verifies that the members in the port channel are mapped to the vPC.
show spanning-tree	Verifies that the following STP parameters are identical when STP is enabled: • BPDU filter • BPDU guard • Cost • Link type • Priority • VLANs (PVRST+)

The following example shows sample output for the **show vpc** command:

```
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 1
Peer status                       : peer link is down

vPC keep-alive status             : Suspended (Destination IP not reachable)
Configuration consistency status  : failed
Per-vlan consistency status       : success

Configuration inconsistency reason: Consistency Check Not Performed
Type-2 inconsistency reason       : Consistency Check Not Performed
vPC role                          : none established
```

```
Number of vPCs configured        : 2
Peer Gateway                     : Enabled
Dual-active excluded VLANs       : -
Graceful Consistency Check       : Disabled (due to peer configuration)
Auto-recovery status             : Disabled

vPC Peer-link status
-------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ ------------------------------------
1    Po10   down   -


vPC status
-------------------------------------------------------
id   Port   Status Consistency Reason                    Active vlans
--   ----   ------ ----------- ------                    ------------
2    Po20   down   failed      Peer-link is down         -



50   Po50   down   failed      Peer-link is down         -
```

Received Type 1 Configuration Element Mismatch

You might have a problem where you cannot bring up a vPC link because of a type 1 configuration element mismatch.

Symptom	Possible Cause	Solution
Received a type 1 configuration element mismatch.	The vPC peer ports or membership ports do not have identical configurations.	Use the **show vpc consistency-parameters interface** command to determine where the configuration mismatch occurs.

This example shows how to display the vPC consistency parameters on a port channel:

```
switch# show vpc consistency-parameters interface po 10
   Legend:
       Type 1 : vPC will be suspended in case of mismatch
Name                        Type  Local Value            Peer Value
-------------               ----  ---------------------- -----------------------
STP Mode                    1     Rapid-PVST             Rapid-PVST
STP Disabled                1     None                   None
STP MST Region Name         1     ""                     ""
STP MST Region Revision     1     0                      0
STP MST Region Instance to  1
 VLAN Mapping
STP Loopguard               1     Disabled               Disabled
STP Bridge Assurance        1     Enabled                Enabled
STP Port Type               1     Normal                 Normal
STP MST Simulate PVST       1     Enabled                Enabled
Allowed VLANs               -     1-10,15-20,30,37,99    1-10,15-20,30,37,99
```

Cannot Enable the vPC Feature

You might receive an error when you enable the vPC feature.

Symptom	Possible Cause	Solution
Cannot enable the vPC feature.	The hardware is incompatible with the vPC.	Use the **show module**command to determine the hardware version of each Ethernet module.

This example shows how to display the module hardware version:

```
switch# show module
Mod Ports Module-Type                         Model              Status
--- ----- ----------------------------------- ------------------ ----------
22   0    Fabric Module                       N9K-C9508-FM       ok
24   0    Fabric Module                       N9K-C9508-FM       ok
26   0    Fabric Module                       N9K-C9508-FM       ok
27   0    Supervisor Module                   N9K-SUP-A          active *
29   0    System Controller                   N9K-SC-A           active
30   0    System Controller                   N9K-SC-A           standby

Mod Sw               Hw
--- ---------------- ------
22  6.1(2)I1(1)      0.4040
24  6.1(2)I1(1)      0.4040
26  6.1(2)I1(1)      0.4040
27  6.1(2)I1(1)      0.4080
29  6.1(2)I1(1)      0.2170
30  6.1(2)I1(1)      0.2170
```

vPCs in Blocking State

vPCs might be in the blocking state because of bridge assurance (BA).

Symptom	Possible Cause	Solution
vPC is in blocking state.	BPDU only sends on a single link of a port channel. If a BA dispute is detected, the entire vPC will be in the blocking state.	Do not enable BA on the vPC.

VLANs on a vPC Moved to Suspend State

VLANs on a vPC might move to the suspend state.

Symptom	Possible Cause	Solution
VLANs on a vPC are moved to the suspend state.	VLANs allowed on the vPC have not been allowed on the vPC peer link.	All VLANs allowed on a vPC must also be allowed on the vPC peer link. Also, we recommend that only vPC VLANs are allowed on the vPC peer link.

Hosts with an HSRP Gateway Cannot Access Beyond Their VLAN

When HSRP is enabled on both vPC peer devices on a VLAN and hosts on that VLAN set the HSRP as their gateway, they might not able to reach anything outside their own VLAN.

Symptom	Possible Cause	Solution
Hosts with an HSRP gateway cannot access beyond their VLAN.	If the host gateway MAC address is mapped to the physical MAC address of any one of the vPC peer devices, packets might get dropped due to the loop prevention mechanism in the vPC.	Map the host gateway's MAC address to the HSRP MAC address and not the physical MAC address of any one of the vPC peer devices. The peer gateway can be a workaround for this scenario. Read the configuration guide for more information about the peer gateway before you implement it.

CHAPTER 7

Troubleshooting VLANs

Troubleshooting VXLAN Issues

The VXLAN data path includes the following paths:

- Multicast encapsulation path—Native Layer 2 packets are encapsulated in VXLAN in the access to network (Layer 2 to Layer 3) direction
- Multicast decapsulation path—Native Layer 2 packets are decapsulated in VXLAN in the network to access (Layer 3 to Layer 2) direction
- Unicast encapsulation path—Native Layer 2 packets are encapsulated in VXLAN in the access to network (Layer 2 to Layer 3) direction
- Unicast decapsulation path—Native Layer 2 packets are decapsulated in VXLAN in the network to access (Layer 3 to Layer 2) direction

Understanding these data paths can help you troubleshoot VXLAN issues.

Caution To troubleshoot VXLAN issues, you need to run Broadcom shell commands. Use these Broadcom shell commands with caution and only under the direct supervision or request of Cisco Support personnel.

Note The Cisco Nexus 9300 Series switches support VXLAN. The Cisco Nexus 9500 Series switches do not.

Packets Dropped in the Multicast Encapsulation Path

Follow these steps if ARP requests or multicast packets are being dropped on the device in the access to network direction.

SUMMARY STEPS

1. Access the Broadcom shell.
2. Check the output of the **stg show** command to see if the ports are in the STP forward state for a given VLAN.
3. Verify if ports are part of the VLAN.
4. Check the output of the **mc show** command to see if the local VLAN ports and encapsulation port are part of the encapsulation flood list.
5. If the output of the **mc show** command is incorrect, exit the Broadcom shell mode, run the following commands, and view the output: **show tech-support pixm**, **show tech-support pixm-all**, and **show tech-support pixmc-all**.

DETAILED STEPS

Procedure

Step 1 Access the Broadcom shell.

Example:

```
switch# bcm-shell module 1
Warning: BCM shell access should be used with caution
Entering bcm shell on module 1
Available Unit Numbers: 0
```

Step 2 Check the output of the **stg show** command to see if the ports are in the STP forward state for a given VLAN.

Example:

```
bcm-shell.0> stg show
STG 6: contains 1 VLAN (3)
  Disable: xe56-xe95
    Block: xe0-xe22,xe24-xe55
  Forward: xe23,hg
```

In this example, VLAN 3 has eth1/24 and uplink tunnel port is eth2/2, so we would expect to see xe23 (1/24) and hg in the output.

Step 3 Verify if ports are part of the VLAN.

Example:

```
bcm-shell.0> vlan show
vlan 3  ports xe23
(0x0000000000000000000000000000000000000000000000000000000100000000), untagged
none (0x0000000000000000000000000000000000
000000000000000000000000000000) MCAST_FLOOD_UNKNOWN
```

In this example, xe23 needs to be part of VLAN 3.

Step 4 Check the output of the **mc show** command to see if the local VLAN ports and encapsulation port are part of the encapsulation flood list.

a) Get the encapsulation flood list.

Example:

```
bcm-shell.0> d chg vfi 3
Private image version: R
VFI.ipipe0[3]:
<VP_1=0xc01,VP_0=0x1803,UUC_INDEX=0x1803,UMC_INDEX=0x1803,RSVD_VP_0=1,BC_INDEX=0x1803>
```

In this example, 0x1803 is the encapsulation flood list.

b) Feed the encapsulation flood list into the **mc show** command.

Example:

```
bcm-shell.0> mc show 0x1803
Group 0xc001803 (VXLAN)
        port hg7, encap id 400053
        port xe23, encap id 400057
```

In this example, hg7 is the uplink tunnel port, and xe23 is the local port in the VLAN.

If the uplink is a port channel, all members of the port channel should appear in the output. If the output includes duplicate entries, there will be a corresponding packet replication.

Step 5 If the output of the **mc show** command is incorrect, exit the Broadcom shell mode, run the following commands, and view the output: **show tech-support pixm**, **show tech-support pixm-all**, and **show tech-support pixmc-all**.

Example:

```
bcm-shell.0> exit
switch# show tech-support pixm
switch# show tech-support pixm-all
switch# show tech-support pixmc-all
```

Packets Dropped in the Multicast Decapsulation Path

Follow these steps if ARP requests or multicast packets are being dropped on the device in the network to access direction.

SUMMARY STEPS

1. Check if the packets were sent to the supervisor and if remote VXLAN tunnel endpoint (VTEP) discovery occurred.
2. If the mpls_entry is present in the hardware, check the vlan_xlate table.
3. If the vlan_xlate table has the correct entry for the multicast DIP, check if the VLAN flood list shows the correct members (that is, the members of the VLAN excluding the encapsulation tunnel port).

DETAILED STEPS

Procedure

Step 1 Check if the packets were sent to the supervisor and if remote VXLAN tunnel endpoint (VTEP) discovery occurred.

a) Check if the remote peer was learned in the software.

Example:

```
switch# show nve peers
Interface          Peer-IP          VNI        Up Time
----------------   ---------------  --------   -----------
nve1               100.100.100.5    10000      00:02:23
```

b) Check if the remote peer was learned in the hardware by checking the mpls_entry table.

Example:

```
switch# bcm-shell module 1
bcm-shell.0> d chg mpls_entry | grep SVP
MPLS_ENTRY.ipipe0[12368]:
<VXLAN_SIP:SVP=0x1751,VXLAN_SIP:SIP=0x66666666,VXLAN_SIP:KEY=0x666666668,VXLAN_SIP:HASH_LSB=0x666
VXLAN_SIP:DATA=0
x1751,VALID=1,KEY_TYPE=8,>
```

c) If the mpls_entry is missing and the source virtual port (SVP) is not present, check if the packets are being sent to the supervisor and check for any IPFIB errors.

Example:

```
bcm-shell.0> show c cpu0
bcm-shell.0> exit
switch# attach module 1
module-1# show system internal ipfib errors
```

Step 2 If the mpls_entry is present in the hardware, check the vlan_xlate table.

Example:

```
module-1# exit
switch# bcm-shell module 1
bcm-shell.0> d chg vlan_xlate | grep 0xe1000003
VLAN_XLATE.ipipe0[8464]:
<VXLAN_DIP:KEY=0x7080000192,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=3,VXLAN_DIP:DIP=0xe1000003,
XLAN_DIP
:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

The vlan_xlate table should have one entry for the multicast destination IP address (DIP) of the packet. This example shows such an when multicast packets are sent to 225.0.0.3.

Step 3 If the vlan_xlate table has the correct entry for the multicast DIP, check if the VLAN flood list shows the correct members (that is, the members of the VLAN excluding the encapsulation tunnel port).

a) Check the VLAN flood list.

Example:

```
bcm-shell.0> d chg vfi 3
Private image version: R
VFI.ipipe0[3]:
<VP_1=0xc01,VP_0=0x1803,UUC_INDEX=0x1803,UMC_INDEX=0x1803,RSVD_VP_0=1,BC_INDEX=0x1803>
```

For the encapsulation flood list of 0x1803, the corresponding decapsulation flood list would be 0x1c03.

b) Check if the local port is part of the decapsulation flood list.

Example:

```
bcm-shell.0> mc show
Group 0xc001c03 (VXLAN)
        port xe23, encap id 400057
```

xe23 must be part of the decapsulation flood list.

c) Make sure the port is in the forwarding state and part of the VLAN.

Example:

```
bcm-shell.0> stg show
bcm-shell.0> vlan show
```

Packets Dropped in the Unicast Encapsulation Path

Unicast Packets Dropped When VTEP Is Reachable Through a Single Next Hop

Follow these steps if unicast packets are being dropped on the device in the access to network direction and VTEP is reachable through a single next hop.

SUMMARY STEPS

1. Check if the remote peer is discovered in the hardware.
2. Get the mapping of the source virtual port (SVP) to the next hop.
3. Get the port number from the next-hop index.
4. Get the mapping from the port number to the physical port on the chip.
5. Get the egress port to next-hop index mapping.
6. Check the tunnel parameters to make sure that the EGR IP tunnel shows the correct local VTEP IP address in the SIP field.
7. Make sure that the tunnel DIP is programmed.

DETAILED STEPS

Procedure

Step 1 Check if the remote peer is discovered in the hardware.

Example:

```
switch# bcm-shell module 1
bcm-shell.0> d chg mpls_entry | grep SVP
MPLS_ENTRY.ipipe0[12368]:
<VXLAN_SIP:SVP=0x1751,VXLAN_SIP:SIP=0x66666666,VXLAN_SIP:KEY=0x666666668,VXLAN_SIP:HASH_LSB=0x666
VXLAN_SIP:DATA=0
x1751,VALID=1,KEY_TYPE=8,>
```

Make sure a valid source IP address (SIP) exists.

In this example, 102.102.102.102 is the remote VTEP IP address.

Step 2 Get the mapping of the source virtual port (SVP) to the next hop.

Example:

```
bcm-shell.0> d chg ing_dvp_table 0x1751
Private image version: R
ING_DVP_TABLE.ipipe0[5969]:
<VP_TYPE=3,NEXT_HOP_INDEX=0x18,NETWORK_PORT=1,ECMP_PTR=0x18,DVP_GROUP_PTR=0x18,>
```

In this example, the next-hop index is 0x18.

Step 3 Get the port number from the next-hop index.

Example:

```
bcm-shell.0> d chg ing_l3_next_hop 0x18
Private image version: R
ING_L3_NEXT_HOP.ipipe0[24]:
<VLAN_ID=0xfff,TGID=0x88,PORT_NUM=8,MTU_SIZE=0x3fff,MODULE_ID=1,L3_OIF=0x1fff,ENTRY_TYPE=2
ENTRY_INFO_UPPER=3,DV
P_RES_INFO=0x7f,>
```

In this example, the port number is 8.

Step 4 Get the mapping from the port number to the physical port on the chip.

Example:

```
bcm-shell.0> phy info
Phy mapping dump:
      port   id0   id1  addr iaddr          name    timeout
  hg0(  1)  600d  8770   1b1   1b1   TSC-A2/31/4     250000
  hg1(  2)  600d  8770    81    81   TSC-A2/00/4     250000
  hg2(  3)  600d  8770   1ad   1ad   TSC-A2/30/4     250000
  hg3(  4)  600d  8770    85    85   TSC-A2/01/4     250000
  hg4(  5)  600d  8770   189   189   TSC-A2/23/4     250000
  hg5(  6)  600d  8770    ad    ad   TSC-A2/08/4     250000
  hg6(  7)  600d  8770   185   185   TSC-A2/22/4     250000
  hg7(  8)  600d  8770    b1    b1   TSC-A2/09/4     250000
  xe0(  9)  600d  84f9     0    89      BCM84848     250000
  xe1( 10)  600d  84f9     1    8a      BCM84848     250000
  xe2( 11)  600d  84f9     2    8b      BCM84848     250000
  xe3( 12)  600d  84f9     3    8c      BCM84848     250000
```

In this example, port number 8 is hg7.

Step 5 Get the egress port to next-hop index mapping.

Example:

```
bcm-shell.0> g chg egr_port_to_nhi_mapping
EGR_PORT_TO_NHI_MAPPING.hg7[2][0x4001808]=0x18: <NEXT_HOP_INDEX=0x18>
```

In this example, next-hop index 0x18 points to hg7.

Step 6 Check the tunnel parameters to make sure that the EGR IP tunnel shows the correct local VTEP IP address in the SIP field.

Example:

```
bcm-shell.0> d chg egr_ip_tunnel
Private image version: R
EGR_IP_TUNNEL.epipe0[1]:
```

```
<TUNNEL_TYPE=0xb,TTL=0xff,SIP=0x65656565,L4_DEST_PORT=0x2118,ENTRY_TYPE=1,DSCP_SEL=1,>
```

In this example, SIP is the local VTEP IP address (101.101.101.101), L4_DEST_PORT is 0x2118 (port 8472), and DSCP_SEL = 1 means that the inner DSCP packet will be copied to the outer DSCP packet.

Step 7 Make sure that the tunnel DIP is programmed.

Example:

```
bcm-shell.0> d chg egr_dvp_attribute 0x1751
Private image version: R
EGR_DVP_ATTRIBUTE.epipe0[5969]:
<VXLAN:TUNNEL_INDEX=1,VXLAN:DVP_IS_NETWORK_PORT=1,VXLAN:DIP=0x66666666,VP_TYPE=2,>
```

Unicast Packets Dropped When VTEP Is Reachable Through an ECMP Path

Follow these steps if unicast packets are being dropped on the device in the access to network direction and VTEP is reachable through an ECMP path.

SUMMARY STEPS

1. Get the ECMP next hop for a given remote peer virtual port (VP).
2. Convert the ECMP_PTR to decimal and add 200000 to get the port number.
3. Get the list of interfaces in the ECMP next-hop set.
4. Find the members of the port channel.
5. Find the physical next-hop interfaces for the given next-hop index.

DETAILED STEPS

Procedure

Step 1 Get the ECMP next hop for a given remote peer virtual port (VP).

Example:

```
bcm-shell.0> d chg ing_dvp_table 0x1751
Private image version: R
ING_DVP_TABLE.ipipe0[5969]:
<VP_TYPE=3,NEXT_HOP_INDEX=0x108,NETWORK_PORT=1,ECMP_PTR=0x108,ECMP=1,DVP_GROUP_PTR=0x108,>
```

In this example, 0x1751 is the VP number for the remote peer IP address derived from using the d chg mpls_entry output.

Note
If the remote VTEP is reachable through an ECMP path, ECMP=1 needs to be present in the output.

Step 2 Convert the ECMP_PTR to decimal and add 200000 to get the port number.

Example:

```
0x108 (264) + 200000 = 200264
```

In this example, the port number is 200264.

Step 3 Get the list of interfaces in the ECMP next-hop set.

Example:

```
bcm-shell.0> d chg l3 multipath show 200264
Multipath Egress Object 200264
Interfaces: 100606 100607 100608
Reference count: 2
bcm-shell.0> l3 egress show | grep 100606
100606  00:22:bd:f5:1a:60 4095 4101     1t   0        -1   no   no
bcm-shell.0> l3 egress show | grep 100607
100607  00:22:bd:f5:1a:60 4095 4102     2t   0        -1   no   no
bcm-shell.0> l3 egress show | grep 100608
100608  00:22:bd:f5:1a:60 4095 4103     3t   0        -1   no   no
```

In this example, the next-hop interfaces are 1t, 2t, and 3t, which are port channels.

Step 4 Find the members of the port channel.

Example:

```
bcm-shell.0> trunk show
Device supports 1072 trunk groups:
  1024 front panel trunks (0..1023), 256 ports/trunk
  48 fabric trunks (1024..1071), 64 ports/trunk
trunk 0: (front panel, 0 ports)
trunk 1: (front panel, 1 ports)=hg6 dlf=any mc=any ipmc=any psc=portflow (0x9)
trunk 2: (front panel, 1 ports)=hg4 dlf=any mc=any ipmc=any psc=portflow (0x9)
trunk 3: (front panel, 1 ports)=hg7 dlf=any mc=any ipmc=any psc=portflow (0x9)
```

Step 5 Find the physical next-hop interfaces for the given next-hop index.

Example:

```
bcm-shell.0> g chg egr_port_to_nhi_mapping
EGR_PORT_TO_NHI_MAPPING.hg4[2][0x4001805]=0x5f7: <NEXT_HOP_INDEX=0x5f7>
EGR_PORT_TO_NHI_MAPPING.hg6[2][0x4001807]=0x9b3: <NEXT_HOP_INDEX=0x9b3>
EGR_PORT_TO_NHI_MAPPING.hg7[2][0x4001808]=0x5f8: <NEXT_HOP_INDEX=0x5f8>
```

In this example, next-hop index 0x5f7 points to hg4, 0x9b3 points to hg6, and 0x5f8 points hg7.

Packets Dropped in the Unicast Decapsulation Path

Follow these steps if unicast packets are being dropped on the device in the network to access direction.

SUMMARY STEPS

1. Check if the packets were sent to the supervisor and if remote VXLAN tunnel endpoint (VTEP) discovery occurred.
2. If the mpls_entry is present in the hardware, check the vlan_xlate table.
3. Check if the unicast DIP entry is present in the vlan_xlate table.
4. Check if the unicast DIP entry is present in the vlan_xlate table.
5. Make sure that the destination MAC address appears in the Layer 2 MAC address table.

DETAILED STEPS

Procedure

Step 1 Check if the packets were sent to the supervisor and if remote VXLAN tunnel endpoint (VTEP) discovery occurred.

a) Check if the remote peer was learned in the software.

Example:

```
switch# show nve peers
Interface          Peer-IP           VNI        Up Time
----------------   ---------------   --------   -----------
nve1               100.100.100.5     10000      00:06:54
```

b) Check if the remote peer was learned in the hardware by checking the mpls_entry table.

Example:

```
switch# bcm-shell module 1
bcm-shell.0> d chg mpls_entry | grep SVP
MPLS_ENTRY.ipipe0[12368]:
<VXLAN_SIP:SVP=0x1751,VXLAN_SIP:SIP=0x66666666,VXLAN_SIP:KEY=0x666666668,VXLAN_SIP:HASH_LSB=0x666
VXLAN_SIP:DATA=0
x1751,VALID=1,KEY_TYPE=8,>
```

c) If the mpls_entry is missing and the source virtual port (SVP) is not present, check if the packets are being sent to the supervisor and check for any IPFIB errors.

Example:

```
bcm-shell.0> show c cpu0
bcm-shell.0> exit
switch# attach module 1
module-1# show system internal ipfib errors
```

Step 2 If the mpls_entry is present in the hardware, check the vlan_xlate table.

Example:

```
module-1# exit
switch# bcm-shell module 1
bcm-shell.0> d chg vlan_xlate | grep 0xe1000003
VLAN_XLATE.ipipe0[8464]:
<VXLAN_DIP:KEY=0x7080000192,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=3,VXLAN_DIP:DIP=0xe1000003,
XLAN_DIP
:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

The vlan_xlate table should have one entry for the multicast destination IP address (DIP) of the packet. This example shows such an when multicast packets are sent to 225.0.0.3.

Step 3 Check if the unicast DIP entry is present in the vlan_xlate table.

Example:

```
bcm-shell.0> d chg vlan_xlate | grep 0x65656565
VLAN_XLATE.ipipe0[14152]:
<VXLAN_DIP:KEY=0x32b2b2b292,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=0x565
```

```
VXLAN_DIP:DIP=0x65656565,VXLAN_DIP:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

If the entry is present, decapsulation should occur.

Step 4 Check if the unicast DIP entry is present in the vlan_xlate table.

Example:

```
bcm-shell.0> d chg vlan_xlate | grep 0x65656565
VLAN_XLATE.ipipe0[14152]:
<VXLAN_DIP:KEY=0x32b2b2b292,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=0x565
VXLAN_DIP:DIP=0x65656565,VXLAN_DIP:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

If the entry is present, decapsulation should occur.

Step 5 Make sure that the destination MAC address appears in the Layer 2 MAC address table.

Example:

```
bcm-shell.0> l2 show
mac=00:00:bb:01:00:03 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:cc:01:00:0a vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:05 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:aa:01:00:0a vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:07 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:cc:01:00:01 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:08 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:aa:01:00:01 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:cc:01:00:07 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:cc:01:00:02 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:04 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:cc:01:00:04 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:02 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:cc:01:00:09 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:09 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:06 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:cc:01:00:06 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:06 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:09 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:bb:01:00:04 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:bb:01:00:02 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:aa:01:00:08 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:07 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:cc:01:00:08 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:01 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:cc:01:00:05 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:03 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:0a vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:cc:01:00:03 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:05 vlan=28772 GPORT=0x80003401Unknown GPORT format
```

If the destination MAC address is present, Layer 2 forwarding occurs. Otherwise, packets will be flooded using the decapsulation flood list.

Understanding Broadcom Shell Tables

This section provides information on Broadcom shell tables with respect to VXLAN.

MPLS Entry Table

The MPLS entry (mpls_entry) table contains the following information:

- The IP address of the remote VTEP (SIP)
- The tunnel encapsulation port (SVP)
- The mapping between the VLAN and the VNID (VFI, VN_ID)

When the SIP entry is missing in the mpls_entry table, the packets are sent to the supervisor for VTEP learning. Once the entry is installed in the hardware, the packets should no longer be sent to the supervisor.

Note Some packets will be dropped during the learning phase because software forwarding is not performed for VXLAN packets.

Note Packets that are sent to the supervisor use the class-default CPU queue. There is not currently a dedicated COPP class for VxLAN.

The following example shows a table where the remote VTEP IP address is 100.100.100.1 and VLAN 100 maps to VNID 10000.

```
bcm-shell.0> d chg mpls_entry
Private image version: R
MPLS_ENTRY.ipipe0[6816]: <VXLAN_SIP:SVP=8,VXLAN_SIP:SIP=0x64646401,VXLAN_SIP:KEY=0x646464018
VXLAN_SIP:HASH_LSB=0x401,VXLAN_SIP:DATA=8,VALID=1,KEY_TYPE=8,>
MPLS_ENTRY.ipipe0[8680]:
<VXLAN_VN_ID:VN_ID=0x2710,VXLAN_VN_ID:VFI=0x64,VXLAN_VN_ID:KEY=0x27109
VXLAN_VN_ID:HASH_LSB=0x710,VXLAN_VN_ID:DATA=0x64,VALID=1,KEY_TYPE=9,>
```

In the output, you are looking for one entry per VLAN-VNID mapping. In this example, VN_ID=0x2710 is the VNID in hexadecimal notation, VFI=0x64 is the mapped VLAN in hexadecimal notation, and 0x64 = 100 maps to 0x2710 VNID 10000.

MAC Address Learning

MAC addresses that are learned in VXLAN VLANs appear as learned over an internal translated VLAN (for example, VLAN 100 appears as VLAN 28772).

GPORT refers to the port or virtual port that the MAC address was learned against. For local MAC addresses, there is mapping between the GPORT# and the front panel port#. Remote MAC addresses should be learned against the SVP that is pointing to the tunnel port.

A miss in this table means flood the packet to local ports in the VLAN and the tunnel port. A hit in this table means forward the packet to the corresponding GPORT. If GPORT is the tunnel port, you need to encapsulate the packet in VXLAN. If GPORT is the local port, then regular Layer 2 learned MAC address forwarding occurs.

Note To get the mapping between the GPORT and the front-panel port number, see the Getting the GPORT to Front-Panel Port Number Mapping, on page 66 section.

Ingress DVP Table

The ingress DVP table maps the virtual port to the next-hop index. It is used in the unicast encapsulation path and is indexed by the virtual port. In the case of ECMP, the ECMP=1 field is needed.

The following example shows that for VP 0x1751 the next-hop index is 0x35.

```
bcm-shell.0> d chg ing_dvp_table 0x1751
Private image version: R
ING_DVP_TABLE.ipipe0[5969]:
<VP_TYPE=3,NEXT_HOP_INDEX=0x35,NETWORK_PORT=1,ECMP_PTR=0x35,DVP_GROUP_PTR=0x35,>
```

Ingress Layer 3 Next Hop

The ingress Layer 3 next hop gives the port number for a given next-hop index. It is used in the unicast encapsulation path. You can use the phy_info to get the mapping between the port number and the actual front-panel port number.

```
bcm-shell.0> d chg ing_l3_next_hop
ING_L3_NEXT_HOP.ipipe0[16]:
<VLAN_ID=0xfff,TGID=0x9f,PORT_NUM=0x1f,MTU_SIZE=0x3fff,MODULE_ID=1,L3_OIF=0x1fff,ENTRY_TYPE=2
ENTRY_INFO_UPPER=3,DVP_RES_INFO=0x7f,>
```

VLAN Translate Table

The VLAN translate table is used in the decapsulation path for both VXLAN multicast and unicast. It contains three types of entries:

- One entry per outer multicast group (multicast DIP)
- One entry for the local VTEP (unicast DIP)
- One entry per VLAN, per port

The following example shows a multicast DIP entry.

```
bcm-shell.0> d chg vlan_xlate | grep 0xe1000003
VLAN_XLATE.ipipe0[8464]:
<VXLAN_DIP:KEY=0x7080000192,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=3
VXLAN_DIP:DIP=0xe1000003,VXLAN_DIP:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

The following example shows a unicast DIP entry.

```
bcm-shell.0> d chg vlan_xlate | grep 0x65656565
VLAN_XLATE.ipipe0[14152]:
<VXLAN_DIP:KEY=0x32b2b2b292,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=0x565
VXLAN_DIP:DIP=0x65656565,VXLAN_DIP:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

The following example shows one entry per VLAN, per port.

```
bcm-shell.0> d chg vlan_xlate | grep VLAN_ID=3
VLAN_XLATE.ipipe0[3216]:
<XLATE:VLAN_ID=3,XLATE:TGID=0xa0,XLATE:SVP_VALID=1,XLATE:SOURCE_VP=0x201,XLATE:SOURCE_FIELD=0xa0
XLATE:PORT_NUM=0x20,XLATE:OVID=3,XLATE:OTAG=3,XLATE:OLD_VLAN_ID=3,XLATE:MPLS_ACTION=1
XLATE:MODULE_ID=1,XLATE:KEY=0x1805024,XLATE:ITAG=3,XLATE:INCOMING_VIDS=3,XLATE:HASH_LSB=3
XLATE:GLP=0xa0,XLATE:DISABLE_VLAN_CHECKS=1,XLATE:DATA=0x100a000000000000000001,VLAN_ID=3
VALID=1,TGID=0xa0,SVP_VALID=1,SOURCE_VP=0x201,SOURCE_TYPE=1,SOURCE_FIELD=0xa0,PORT_NUM=0x20,OVID=3
OTAG=3,OLD_VLAN_ID=3,MPLS_ACTION=1,MODULE_ID=1,KEY_TYPE=4,KEY=0x1805024,ITAG=3,INCOMING_VIDS=3
HASH_LSB=3,GLP=0xa0,DISABLE_VLAN_CHECKS=1,DATA=0x100a000000000000000001>
```

EGR Port to NHI Mapping

EGR port to NHI mapping maps the next-hop index to the egress port. It is used in the unicast encapsulation path.

```
bcm-shell.0> g chg egr_port_to_nhi_mapping
EGR_PORT_TO_NHI_MAPPING.hg7[2][0x4001808]=0x36: <NEXT_HOP_INDEX=0x36>
```

VLAN Flood Index Table

The VLAN flood index (VFI) table shows the BC/UUC/UMC index for a given VLAN or VFI. The flood index can be used in the output of the **mc show** command to find the members of the VLAN, including the tunnel encapsulation port.

The following example shows how to get the port number.

```
bcm-shell.0> d chg vfi 3
Private image version: R
VFI.ipipe0[3]:
<VP_1=0xc01,VP_0=0x1803,UUC_INDEX=0x1803,UMC_INDEX=0x1803,RSVD_VP_0=1,BC_INDEX=0x1803>
```

The following example shows how to feed this port number into the phy_info to get the front-panel port number.

```
bcm-shell.0> d chg ing_l3_next_hop
ING_L3_NEXT_HOP.ipipe0[16]:
<VLAN_ID=0xfff,TGID=0x9f,PORT_NUM=0x1f,MTU_SIZE=0x3fff,MODULE_ID=1,L3_OIF=0x1fff,ENTRY_TYPE=2
ENTRY_INFO_UPPER=3,DVP_RES_INFO=0x7f,

bcm-shell.0> phy info
Phy mapping dump:
      port   id0   id1  addr iaddr                    name    timeout
  hg0(  1)  600d  8770   1b1   1b1            TSC-A0/31/4     250000
  hg1(  2)  600d  8770    81    81            TSC-A0/00/4     250000
  hg2(  3)  600d  8770   1ad   1ad            TSC-A0/30/4     250000
  hg3(  4)  600d  8770    85    85            TSC-A0/01/4     250000
  hg4(  5)  600d  8770   1a9   1a9            TSC-A0/29/4     250000
  hg5(  6)  600d  8770    89    89            TSC-A0/02/4     250000
  hg6(  7)  600d  8770   195   195            TSC-A0/26/4     250000
  hg7(  8)  600d  8770    a1    a1            TSC-A0/05/4     250000
  hg8(  9)  600d  8770   191   191            TSC-A0/25/4     250000
```

The following example shows the decapsulation route:

```
bcm-shell.0> d chg vlan_xlate
Private image version: R
VLAN_XLATE.ipipe0[768]:
<VXLAN_DIP:NETWORK_RECEIVERS_PRESENT=1,VXLAN_DIP:KEY=0x7080000092,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1
```

```
VXLAN_DIP:HASH_LSB=1,VXLAN_DIP:DIP=0xe1000001,VXLAN_DIP:DATA=0x400001,VALID=1,KEY_TYPE=0x12,>
VLAN_XLATE.ipipe0[1472]:
<VXLAN_DIP:KEY=0x3232320112,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=0x402
VXLAN_DIP:DIP=0x64646402,VXLAN_DIP:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

Note The NETWORK_RECEIVERS_PRESENT must be set to 0.

Getting the GPORT to Front-Panel Port Number Mapping

Follow these steps to get the mapping between the GPORT and the front-panel port number.

SUMMARY STEPS

1. Use this formula to get the local target logic (LTL) from the GPORT#: LTL# = 0x10000 - 512 + GPORT#
2. Get the ifindex for a given LTL.
3. Get the ifindex to the front-panel port.
4. Display the GPORT to front-panel port number mapping.

DETAILED STEPS

Procedure

Step 1 Use this formula to get the local target logic (LTL) from the GPORT#: LTL# = 0x10000 - 512 + GPORT#

For a GPORT of 0x201, the LTL is 0x10000 + 0x201 (513) - 0x200 (512) = 0x10001.

Step 2 Get the ifindex for a given LTL.

Example:

```
switch# attach module 1
module-1# show system internal pixmc info sdb ltl 0x10001
```

Step 3 Get the ifindex to the front-panel port.

Example:

```
module-1# exit
switch# show int snmp-ifindex | grep 0x1a002e00
Eth1/24    436219392  (0x1a002e00)
```

Step 4 Display the GPORT to front-panel port number mapping.

Example:

```
switch# bcm-shell module 1
bcm-shell.0> l2 show
mac=00:00:00:00:00:00 vlan=0 GPORT=0xc000000 Trunk=0^M
mac=00:00:bb:01:00:03 vlan=28772 GPORT=0x80001751Unknown GPORT format ^M
mac=00:00:cc:01:00:0a vlan=28772 GPORT=0x80000201Unknown GPORT format ^M
```

```
mac=00:00:bb:01:00:05 vlan=28772 GPORT=0x80001751Unknown GPORT format ^M
mac=00:00:aa:01:00:0a vlan=28772 GPORT=0x80000202Unknown GPORT format ^M
```

In this example, MAC address 00:00:bb:01:00:05 is learned over the tunnel, so a GPORT of 0x1751 corresponds to the tunnel SVP. MAC address 00:00:aa:01:00:0a is learned locally, so a GPORT of 0x202 corresponds to the front-panel port.

Finding Which Interface Traffic Will Use for an Egress Port

The following example shows how to find the interface that traffic will use for a given egress port.

```
switch# show system internal ethpm info interface ethernet 2/3 | grep ns_pid
  IF_STATIC_INFO:
port_name=Ethernet2/3,if_index:0x1a006400,ltl=2543,slot=0,nxos_port=50,dmod=1,dpid=9,unit=0
queue=2064,xbar_unitbmp=0x0
ns_pid=8

- dpid=9 is higig8

switch# bcm-shell module 1
bcm-shell.0> g chg egr_port_to_nhi_mapping
EGR_PORT_TO_NHI_MAPPING.hg7[2][0x4001808]=0x36: <NEXT_HOP_INDEX=0x36>
bcm-shell.0> d chg egr_l3_next_hop 0x36
Private image version: R
EGR_L3_NEXT_HOP.epipe0[54]:
<OVID=0x65,MAC_ADDRESS=0x60735cde6e41,L3MC:VNTAG_P=1,L3MC:VNTAG_FORCE_L=1,L3MC:VNTAG_DST_VIF=0x18
L3MC:RSVD_DVP=1,L3MC:INTF_NUM=0x1065,L3MC:FLEX_CTR_POOL_NUMBER=3,L3MC:FLEX_CTR_OFFSET_MODE=3
L3MC:FLEX_CTR_BASE_COUNTER_IDX=0xe41,L3MC:ETAG_PCP_DE_SOURCE=3,L3MC:ETAG_PCP=1
L3MC:ETAG_DOT1P_MAPPING_PTR=1,L3MC:DVP=0x2b9b,L3:OVID=0x65,L3:MAC_ADDRESS=0x60735cde6e41
L3:IVID=0xc83,L3:INTF_NUM=0x1065,IVID=0xc83,INTF_NUM=0x1065,>
```

Finding the Flood List for a VLAN

The following example shows how to find the flood list for a given VLAN.

```
bcm-shell.0> d chg vfi 3
Private image version: R
VFI.ipipe0[3]:
<VP_1=0xc01,VP_0=0x1803,UUC_INDEX=0x1803,UMC_INDEX=0x1803,RSVD_VP_0=1,BC_INDEX=0x1803>
```

Determining if the Encapsulation Port is Part of the Flood List

The following example shows how to determine if the encapsulation port is part of the flood list in the access to network direction.

```
bcm-shell.0> mc show 0x1803
Group 0xc001803 (VXLAN)
        port hg7, encap id 400053
        port xe23, encap id 400057
```

CHAPTER 8

Troubleshooting STP

-

About Troubleshooting STP

STP provides a loop-free network at the Layer 2 level. Layer 2 LAN ports send and receive STP frames at regular intervals. Network devices do not forward these frames but use the frames to construct a loop-free path. For more information on Layer 2, see the *Cisco Nexus 9000 Series Layer 2 Configuration Guide*.

Initial Troubleshooting STP Checklist

Troubleshooting an STP problem involves gathering information about the configuration and connectivity of individual devices and the entire network.

Begin troubleshooting STP issues by checking the following issues first:

Checklist	Done
Verify the type of spanning tree configured on your device.	
Verify the network topology including all interconnected ports and switches. Identify all redundant paths on the network and verify that the redundant paths are blocking.	
Use the **show spanning-tree summary totals** command to verify that the total number of logical interfaces in the Active state are less than the maximum allowed. For information on these limits, see the *Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide*.	
Verify the primary and secondary root bridge and any configured Cisco extensions.	

Use the following commands to view STP configuration and operational details:

- **show running-config spanning-tree**
- **show spanning-tree summary**
- **show spanning-tree detail**
- **show spanning-tree bridge**
- **show spanning-tree mst**
- **show spanning-tree mst configuration**
- **show spanning-tree interface** *interface-type slot*/*port* [**detail**]
- **show tech-support stp**
- **show spanning-tree vlan**

Use the **show spanning-tree blockedports** command to display the ports that are blocked by STP.

Use the **show mac address-table dynamic vlan** command to determine if learning or aging occurs at each node.

Troubleshooting STP Data Loops

Data loops are a common problem in STP networks. Some of the symptoms of a data loop are as follows:

- High link utilization, up to 100 percent
- High CPU and backplane traffic utilization
- Constant MAC address relearning and flapping
- Excessive output drops on an interface

When the l2fm logging level is greater than or equal to 4, the switch logs occurrences of host MAC address flapping to help you locate STP data loops. If it detects a MAC address move within less than 1 second and if 10 consecutive moves occur, the switch disables learning on the VLAN for one of the ports between which the MAC address is moving. Learning is disabled for 120 seconds and reenabled automatically. Syslogs are generated while learning is disabled and enabled. You can configure the logging level using the **logging level l2fm** *log-level* command.

SUMMARY STEPS

1. switch# **show interface** *interface-type slot*/*port* **include rate**
2. switch(config)# **interface** *interface-type slot*/*port*
3. switch(config-if)# **shutdown**
4. switch(config-if)# **show spanning-tree vlan** *vlan-id*
5. (Optional) switch(config-if)# **show spanning-tree interface** *interface-type slot*/*port* **detail**
6. (Optional) switch(config-if)# **show system internal pktmgr interface** *interface-type slot*/*port*
7. (Optional) switch(config-if)# **show system internal pktmgr client** *client-id*
8. (Optional) switch(config-if)# **show interface counters errors**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	switch# **show interface** *interface-type slot*/*port* **include rate**
	Example:
	`switch# show interface ethernet 2/1 include rate`
	` 1 minute input rate 19968 bits/sec, 0 packets/sec`
	` 1 minute output rate 3952023552 bits/sec, 957312`
	` packets/sec`
		Identifies the ports involved in the loop by looking at the interfaces with high link utilization.

Step 2	switch(config)# **interface** *interface-type slot*/*port* **Example:** `switch(config)# interface ethernet 2/1`	Configures the interface type and location.
Step 3	switch(config-if)# **shutdown** **Example:** `switch(config-if)# shutdown`	Shuts down or disconnects the affected ports. After disconnecting the affected ports, locate every switch in the redundant paths using your network topology diagram.

Step 4	switch(config-if)# **show spanning-tree vlan** *vlan-id*
	Example:
	`switch(config-if)# show spanning-tree vlan 9`
	`VLAN0009`
	` Spanning tree enabled protocol rstp`
	`  Root ID   Priority   32777''`
	`            Address    0018.bad7.db15''`
	`            Cost       4`
	`...`
		Verifies that the switch lists the same STP root bridge as the other nonaffected switches.

Step 5	(Optional) switch(config-if)# **show spanning-tree interface** *interface-type slot*/*port* **detail**
	Example:
	`switch(config-if)# show spanning-tree interface`
	`ethernet 3/1 detail`
	`Port 385 (Ethernet3/1) of VLAN0001 is root`
	`forwarding`
	`   Port path cost 4, Port priority 128, Port`
	`Identifier 128.385`
	`   Designated root has priority 32769, address`
	`0018.bad7.db15`
	`   Designated bridge has priority 32769, address`
	`0018.bad7.db15`
	`   Designated port id is 128.385, designated path`
	` cost 0`
	`   Timers: message age 16, forward delay 0, hold`
	`0`
		Verifies that the root port and alternate ports are regularly receiving BPDUs.

	Command or Action	Purpose

	`   Number of transitions to forwarding state: 1`
	`   The port type is network by default`
	`   Link type is point-to-point by default`
	`   BPDU: sent 1265,  received 1269`	

Step 6	(Optional) switch(config-if)# **show system internal pktmgr interface** *interface-type slot*/*port*
	Example:
	`switch(config-if)# show system internal pktmgr`
	`interface ethernet 3/1`
	`Ethernet3/1, ordinal: 36`
	`  SUP-traffic statistics: (sent/received)`
	`    Packets: 120210 / 15812`
	`    Bytes: 8166401 / 1083056`
	`    Instant packet rate: 5 pps / 5 pps`
	`    Average packet rates(1min/5min/15min/EWMA):`
	`    Packet statistics:`
	`      Tx: Unicast 0, Multicast 120210`
	`          Broadcast 0`
	`      Rx: Unicast 0, '' Multicast 15812''`
	`          Broadcast 0`
		Checks if the BPDUs are received by the internal packet manager.

Step 7	(Optional) switch(config-if)# **show system internal pktmgr client** *client-id*
	Example:
	`switch(config-if)# show system internal pktmgr`
	`client 303`
	`Client uuid: 303, 2 filters`
	`   Filter 0: EthType 0x4242, Dmac 0180.c200.0000`
	`   Filter 0: EthType 0x010b, Snap 267, Dmac`
	`0100.0ccc.cccd`
	`   Options: TO 0, Flags 0x1, AppId 0, Epid 0`
	`   Ctrl SAP: 171, Data SAP 177 (1)`
	`   Rx: 28356632, Drop: 0, Tx: 35498365, Drop: 0`
		Checks if the BPDUs are received by the client.

Step 8	(Optional) switch(config-if)# **show interface counters errors**
	Example:
	`switch(config-if)# show interface counters errors`
	
	`--------------------------------------------------------`
	`Port Align-Err FCS-Err Xmit-Err Rcv-Err UnderSize`
	` OutDiscards`
	`--------------------------------------------------------`
	`mgmt0  --       --       --       --      --`
	`   --`
	`Eth1/1  0        0        0        0       0`
	`   0`
	`Eth1/2  0        0        0        0       0`
	`   0`
	`Eth1/3  0        0        0        0       0`
	`   0`
	`Eth1/4  0        0        0        0       0`
	`   0`
		Checks the hardware packet statistic (error drop) counters.

	Command or Action	Purpose

	`Eth1/5  0        0        0        0       0`
	`   0`
	`Eth1/6  0        0        0        0       0`
	`   0`
	`Eth1/7  0        0        0        0       0`
	`   0`
	`Eth1/8  0        0        0        0       0`
	`   0`	

Example

This example shows that the designated port is regularly sending BPDUs:

```
switch# show spanning-tree interface ethernet 3/1 detail
Port 385 (Ethernet3/1) of VLAN0001 is root forwarding
   Port path cost 4, Port priority 128, Port Identifier 128.385
   Designated root has priority 32769, address 0018.bad7.db15
   Designated bridge has priority 32769, address 0018.bad7.db15
   Designated port id is 128.385, designated path cost 0
   Timers: message age 16, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   The port type is network by default
   Link type is point-to-point by default
   BPDU: sent 1265, received 1269
```

This example shows how to check if BPDUs are transmitted by the packet manager:

```
switch# show system internal pktmgr interface ethernet 3/1
Ethernet3/1, ordinal: 36
   SUP-traffic statistics: (sent/received)
      Packets: 120210 / 15812
      Bytes: 8166401 / 1083056
      Instant packet rate: 5 pps / 5 pps
      Average packet rates(1min/5min/15min/EWMA):
      Packet statistics:
        Tx: Unicast 0, M'' ulticast 120210''
            Broadcast 0
        Rx: Unicast 0, Multicast 15812
            Broadcast 0

switch# show system internal pktmgr client 303
Client uuid: 303, 2 filters
   Filter 0: EthType 0x4242, Dmac 0180.c200.0000
   Filter 0: EthType 0x010b, Snap 267, Dmac 0100.0ccc.cccd
   Options: TO 0, Flags 0x1, AppId 0, Epid 0
   Ctrl SAP: 171, Data SAP 177 (1)
   Rx: 28356632, Drop: 0, Tx: 35498365, Drop: 0
```

This example shows how to check the hardware packet statistic counters for a possible BPDU error drop:

```
switch# show interface counters errors
--------------------------------------------------------------
Port Align-Err FCS-Err Xmit-Err Rcv-Err UnderSize OutDiscards
--------------------------------------------------------------
mgmt0  --       --       --       --      --       --
Eth1/1  0        0        0        0       0        0
Eth1/2  0        0        0        0       0        0
```

```
Eth1/3  0        0        0        0        0        0
Eth1/4  0        0        0        0        0        0
Eth1/5  0        0        0        0        0        0
Eth1/6  0        0        0        0        0        0
Eth1/7  0        0        0        0        0        0
Eth1/8  0        0        0        0        0        0
```

Troubleshooting Excessive Packet Flooding

Unstable STP topology changes can trigger excessive packet flooding in your STP network. With Rapid STP or Multiple STP (MST), a change of the port's state to forwarding, as well as the role change from designated to root, can trigger a topology change. Rapid STP immediately flushes the Layer 2 forwarding table. 802.1D shortens the aging time. The immediate flushing of the forwarding table restores connectivity faster but causes more flooding.

In a stable topology, a topology change should not trigger excessive flooding. Link flaps can cause a topology change, so continuous link flaps can cause repetitive topology changes and flooding. Flooding slows the network performance and can cause packet drops on an interface.

SUMMARY STEPS

1. switch# **show spanning-tree vlan** *vlan-id* **detail**
2. switch# **show spanning-tree vlan** *vlan-id* **detail**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	switch# **show spanning-tree vlan** *vlan-id* **detail**
	Example:
	`switch# show spanning-tree vlan 9 detail`
	`VLAN0009 is executing the rstp compatible Spanning`
	` Tree protocol`
	`   Bridge Identifier has priority 32768, sysid 9,`
	` address 0018.bad8.27ad`
	`   Configured hello time 2, max age 20, forward`
	`delay 15`
	`   Current root has priority 32777, address`
	`0018.bad7.db15`
	`   Root port is 385 (Ethernet3/1), cost of root`
	`path is 4`
	`   Topology change flag not set, detected flag not`
	` set`
	`   '' Number of topology changes 8 last change`
	`occurred 1:32:11 ago''`
	`      '' from Ethernet3/1''`
	`   Times: hold 1, topology change 35, notification`
	` 2`
	`...`
		Determines the source of the excessive topology change.

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td><b>Step 2</b></td><td>switch# <b>show spanning-tree vlan</b> <i>vlan-id</i> <b>detail</b>
<b>Example:</b>
<pre>
switch# show spanning-tree vlan 9 detail
VLAN0009 is executing the rstp compatible Spanning
 Tree protocol
   Bridge Identifier has priority 32768, sysid 9,
 address 0018.bad8.27ad
   Configured hello time 2, max age 20, forward
delay 15
   Current root has priority 32777, address
0018.bad7.db15
   Root port is 385 (Ethernet3/1), cost of root
path is 4
   Topology change flag not set, detected flag not
 set
   Number of topology changes 8 last change
occurred 1:32:11 ago
      '' from Ethernet3/1''
   Times: hold 1, topology change 35, notification
 2
   ...
</pre></td><td>Determines the interface where the topology change occurred.
Repeat this step on devices connected to the interface until you can isolate the device that originated the topology change.
Check for link flaps on the interfaces on this device.</td></tr>
</table>

Troubleshooting Convergence Time Issues

STP convergence can take longer than expected or result in an unexpected final network topology.

To troubleshoot convergence issues, check the following issues:

- Errors in the documented network topology diagram.
- Misconfiguration of the timers; diameter; Cisco extension features such as bridge assurance, root guard, and BPDU guard; and so on.
- Overloaded switch CPU during convergence that exceeds the recommended logical port (port-vlan) limit.
- Software defects that affect STP.

Securing the Network Against Forwarding Loops

To handle the inability of STP to deal correctly with certain failures, Cisco has developed a number of features and enhancements to protect the networks against forwarding loops.

Troubleshooting STP helps to isolate and find the cause for a particular failure, while the implementation of these enhancements is the only way to secure the network against forwarding loops.

Before you begin

- Enable the Cisco-proprietary Unidirectional Link Detection (UDLD) protocol on all the switch-to-switch links. For information, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.
- Set up the bridge assurance feature by configuring all the switch-to-switch links as the spanning tree network port type.

Note You should enable the bridge assurance feature on both sides of the links. Otherwise, Cisco NX-OS will put the port in the blocked state because of a bridge assurance inconsistency.

- Set up all the end-station ports as a spanning tree edge port type.

  You must set up the STP edge port to limit the amount of topology change notices and subsequent flooding that can affect the performance of the network. Use this command only with ports that connect to end stations. Otherwise, an accidental topology loop can cause a data-packet loop and disrupt the device and network operation.

- Enable the Link Aggregation Control Protocol (LACP) for port channels to avoid any port-channel misconfiguration issues. For information, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.

  Do not disable autonegotiation on the switch-to-switch links. Autonegotiation mechanisms can convey remote fault information, which is the quickest way to detect failures at the remote side. If failures are detected at the remote side, the local side brings down the link even if the link is still receiving pulses.

Caution Be careful when you change STP timers. STP timers are dependent on each other, and changes can impact the entire network.

SUMMARY STEPS

1. (Optional) switch(config)# **spanning-tree loopguard default**
2. switch(config)# **spanning-tree bpduguard enable**
3. switch(config)# **vlan** *vlan-range*
4. switch(config)# **spanning-tree vlan** *vlan-range* **root primary**
5. switch(config)# **spanning-tree vlan** *vlan-range* **root secondary**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) switch(config)# **spanning-tree loopguard default** **Example:** `switch(config)# spanning-tree loopguard default`	Secures the network STP perimeter with root guard. Root guard and BPDU guard allow you to secure STP against influence from the outside.
Step 2	switch(config)# **spanning-tree bpduguard enable** **Example:** `switch(config)# spanning-tree bpduguard enable`	Enables BPDU guard on STP edge ports to prevent STP from being affected by unauthorized network devices (such as hubs, switches, and bridging routers) that are connected to the ports.

	Command or Action	Purpose
		Root guard prevents STP from outside influences. BPDU guard shuts down the ports that are receiving any BPDUs (not only superior BPDUs). **Note** Short-living loops are not prevented by root guard or BPDU guard if two STP edge ports are connected directly or through the hub.
Step 3	switch(config)# **vlan** *vlan-range* **Example:** `switch(config)# vlan 9`	Configures separate VLANs and avoids user traffic on the management VLAN. The management VLAN is contained to a building block, not the entire network.
Step 4	switch(config)# **spanning-tree vlan** *vlan-range* **root primary** **Example:** `switch(config)# spanning-tree vlan 9 root primary`	Configures a predictable STP root.
Step 5	switch(config)# **spanning-tree vlan** *vlan-range* **root secondary** **Example:** `switch(config)# spanning-tree vlan 12 root secondary`	Configures a predictable backup STP root placement. You must configure the STP root and backup STP root so that convergence occurs in a predictable way and builds optimal topology in every scenario. Do not leave the STP priority at the default value.

CHAPTER 9

Troubleshooting Routing

About Troubleshooting Routing Issues

Layer 3 routing involves determining optimal routing paths and packet switching. You can use routing algorithms to calculate the optimal path from the router to a destination. This calculation depends on the algorithm selected, route metrics, and other considerations such as load balancing and alternate path discovery.

Cisco NX-OS supports multiple virtual routing and forwarding (VRF) instances and multiple routing information bases (RIBs) to support multiple address domains. Each VRF is associated with a RIB, and this information is collected by the Forwarding Information Base (FIB).

See the following documents for more information on routing:

- *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*
- *Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide*

Initial Troubleshooting Routing Checklist

You can troubleshoot routing issues by checking these items first:

Checklist	Done
Verify that the routing protocol is enabled.	
Verify that the address family is configured if necessary.	
Verify that you have configured the correct VRF for your routing protocol.	

Use the following commands to display routing information:

- **show ip arp**

- **show ip traffic**
- **show ip static-route**
- **show ip client**
- **show ip fib**
- **show ip process**
- **show ip route**
- **show vrf**
- **show vrf interface**

Troubleshooting Routing

SUMMARY STEPS

1. switch# **show ospf**
2. switch# **show running-config eigrp all**
3. switch# **show running-config eigrp**
4. switch# **show processes memory | include isis**
5. switch# **show ip client pim**
6. switch# **show ip interface** *loopback-interface*
7. switch# **show vrf interface** *loopback -interface*
8. switch# **show routing unicast clients**
9. switch# **show forwarding distribution multicast client**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	switch# **show ospf**
	Example:
	`switch# show ospf`
	`                  ^`
	`% invalid command detected at '^' marker.`
		Verifies that the routing protocol is enabled.
		If the feature is not enabled, Cisco NX-OS reports that the command is invalid.

Step 2	switch# **show running-config eigrp all** **Example:** `switch# show running-config eigrp all`	Verifies the configuration for this routing protocol.
Step 3	switch# **show running-config eigrp** **Example:**	Verifies the VRF configuration for this routing protocol.

	Command or Action	Purpose

	`switch# show running-config eigrp`
	`version 6.1(2)I1(1)`
	`feature eigrp`
	`router eigrp 99`
	`  address-family ipv4 unicast`
	`    router-id 192.0.2.1`
	`  vrf red`
	`    stub`	

Step 4	switch# **show processes memory | include isis**
	Example:
	`switch# show processes memory | include isis`
	` 8913   9293824  bffff1d0/bffff0d0  isis`
	`32243   8609792  bfffe0c0/bfffdfc0  isis`
		Checks the memory utilization for this routing protocol.

Step 5	switch# **show ip client pim**
	Example:
	`switch# show ip client pim`
	`  Client: pim, uuid: 284, pid: 3839, extended pid:`
	` 3839`
	`  Protocol: 103, client-index: 10, routing VRF id:`
	` 255`
	`   Data MTS-SAP: 1519`
	`   Data messages, send successful: 2135, failed: 0`
		Verifies that the routing protocol is receiving packets.

Step 6	switch# **show ip interface** *loopback-interface*
	Example:
	`switch# show ip interface loopback0`
	`loopback0, Interface status:`
	`protocol-up/link-up/admin-up, iod: 36,`
	`Context:"default"`
	`  IP address: 1.0.0.1, IP subnet: 1.0.0.0/24`
	`  ...`
	`  IP multicast groups locally joined:`
	`      224.0.0.2  224.0.0.1  224.0.0.13`
	`  ...`
		Verifies that the routing protocol is enabled on an interface.

Step 7	switch# **show vrf interface** *loopback -interface*
	Example:
	`switch# show vrf interface loopback 99`
	`Interface                    VRF-Name`
	`         VRF-ID`
	`loopback99                   default`
	`              1`
		Verifies that the interface is in the correct VRF.

Step 8	switch# **show routing unicast clients** **Example:** `switch# show routing unicast clients`	Verifies that the routing protocol is registered with the RIB.

Step 9	switch# **show forwarding distribution multicast client**
	Example:
	`switch# show forwarding distribution multicast`
	`client`
		Verifies that the RIB is interacting with the forwarding plane.

	Command or Action	Purpose
	Number of Clients Registered: 3 Client-name Client-id Shared Memory Name igmp 1 N/A mrib 2 /procket/shm/mrib-mfdm	

Example

This example shows how to display the EIGRP routing protocol configuration:

```
switch# show running-config eigrp all
version 6.1(2)I1(1)
feature eigrp
router eigrp 99
log-neighbor-warnings
  log-neighbor-changes
  log-adjacency-changes
  graceful-restart
  nsf
  timers nsf signal 20
  distance 90 170
  metric weights 0 1 0 1 0 0
  metric maximum-hops 100
  default-metric 100000 100 255 1 1500
  maximum-paths 16
  address-family ipv4 unicast
    log-neighbor-warnings
    log-neighbor-changes
    log-adjacency-changes
    graceful-restart
    router-id 192.0.2.1
    nsf
    timers nsf signal 20
    distance 90 170
    metric weights 0 1 0 1 0 0
    metric maximum-hops 100
    default-metric 100000 100 255 1 1500
    maximum-paths 16
```

This example shows how to display that the unicast routing protocol is registered with the RIB:

```
switch# show routing unicast clients
CLIENT: am
 index mask: 0x00000002
 epid: 3908      MTS SAP: 252       MRU cache hits/misses:        2/1
 Routing Instances:
  VRF: management        table: base
 Messages received:
  Register          : 1      Add-route         : 2      Delete-route      : 1
 Messages sent:
  Add-route-ack     : 2      Delete-route-ack  : 1
CLIENT: rpm
 index mask: 0x00000004
 epid: 4132      MTS SAP: 348       MRU cache hits/misses:        0/0
 Messages received:
  Register          : 1
 Messages sent:
```

```
...
CLIENT: eigrp-99
 index mask: 0x00002000
 epid: 3148      MTS SAP: 63775     MRU cache hits/misses:        0/1
 Routing Instances:
  VRF: default           table: base               notifiers: self
 Messages received:
  Register          : 1      Delete-all-routes : 1
 Messages sent:
...
```

Troubleshooting Policy-Based Routing

- Make sure the ACLs match the incoming traffic.
- Make sure the route is available:
  - For IP network routes, use the **show ip route** command to make sure the IP network route is available for the next hop specified in the **set ip next-hop** command.
  - For IP host routes, use the **show ip arp** command to make sure the IP host route is available for the next hop specified in the **set ip next-hop** command.
  - For IPv6 network routes, use the **show ipv6 route** command to make sure the IPv6 network route is available for the next hop specified in the **set ipv6 next-hop** command.
  - For IPv6 host routes, use the **show ipv6 neighbor** command to make sure the IPv6 host route is available for the next hop specified in the **set ipv6 next-hop** command.
- Make sure the policy is active in the system (using the **show ip policy** command).
- Check the statistics for the entry (using the **show route-map** *map-name* **pbr-statistics** command).

Troubleshoot Dynamic Load Balancing

Consistency checker can be used to troubleshoot the routes using DLB ECMP as follows:

- Global Consistency Checker
  - **test consistency-checker forwarding ipv4 unicast**
  - **show consistency-checker forwarding ipv4 unicast**

  Sample output

```
Leaf1# test consistency-checker forwarding ipv4 unicast
Consistency check started.
Leaf1#
Leaf1#
Leaf1# show consistency-checker forwarding ipv4 unicast
IPV4 Consistency check : table_id(0x1)
Execution time : 28 ms ()
No inconsistent adjacencies.
No inconsistent routes.
Consistency-Checker: PASS for ALL
```

- Single Route Consistency Checker
  - **show consistency-checker forwarding single-route ipv4** *ipv4 address* **vrf** *vrf*

  Sample output

  ```
  Leaf1# show consistency-checker forwarding single-route ipv4 64.60.60.0/24 vrf default


  Consistency checker passed for 64.60.60.0/24
  Leaf1#
  ```

CHAPTER 10

Troubleshooting Memory

About Troubleshooting Memory

Dynamic random access memory (DRAM) is a limited resource on all platforms and must be controlled or monitored to ensure utilization is kept in check.

Cisco NX-OS uses memory in the following three ways:

- **Page cache**—When you access files from persistent storage (CompactFlash), the kernel reads the data into the page cache, which means that when you access the data in the future, you can avoid the slow access times that are associated with disk storage. Cached pages can be released by the kernel if the memory is needed by other processes. Some file systems (tmpfs) exist purely in the page cache (for example, /dev/sh, /var/sysmgr, /var/tmp), which means that there is no persistent storage of this data and that when the data is removed from the page cache, it cannot be recovered. tmpfs-cached files release page-cached pages only when they are deleted.

- **Kernel**—The kernel needs memory to store its own text, data, and Kernel Loadable Modules (KLMs). KLMs are pieces of code that are loaded into the kernel (as opposed to being a separate user process). An example of kernel memory usage is when an inband port driver allocates memory to receive packets.

- **User processes**—This memory is used by Cisco NX-OS or Linux processes that are not integrated in the kernel (such as text, stack, heap, and so on).

When you are troubleshooting high memory utilization, you must first determine what type of utilization is high (process, page cache, or kernel). Once you have identified the type of utilization, you can use additional troubleshooting commands to help you figure out which component is causing this behavior.

General/High Level Assessment of Platform Memory Utilization

You can assess the overall level of memory utilization on the platform by using two basic CLI commands: **show system resources**and **show processes memory**.

Note From these command outputs, you might be able to tell that platform utilization is higher than normal/expected, but you will not be able to tell what type of memory usage is high.

Note If the *show system resources* command output shows a decline in the free memory, it may be because of Linux kernel caching. Whenever the system requires more memory, Linux kernel will release cached memory. The *show system internal kernel meminfo* command displays cached memory in the system.

The **show system resources** command displays platform memory statistics.

```
switch# show system resources
Load average:   1 minute: 0.70   5 minutes: 0.89   15 minutes: 0.88
Processes   :   805 total, 1 running
CPU states  :   7.06% user,   5.49% kernel,   87.43% idle
                  CPU0 states  :   9.67% user,   6.45% kernel,   83.87% idle
                  CPU1 states  :   10.41% user,   7.29% kernel,   82.29% idle
                  CPU2 states  :   5.20% user,   4.16% kernel,   90.62% idle
                  CPU3 states  :   5.15% user,   2.06% kernel,   92.78% idle
Memory usage:   16399900K total,   6557936K used,   9841964K free
Kernel vmalloc:   36168240K total,   18446744039385981489K free     >>>>>>>>>>>>
Kernel buffers:   10860132K Used>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>.>>>>>>>
Kernel cached :   120072K Used >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> seeing these extra
logs
Current memory status: OK
```

```
switch# show system resources
Load average: 1 minute: 0.43 5 minutes: 0.30 15 minutes: 0.28
Processes : 884 total, 1 running
CPU states : 2.0% user, 1.5% kernel, 96.5% idle
Memory usage: 4135780K total, 3423272K used, 712508K free
0K buffers, 1739356K cache
```

Note This output is derived from the Linux memory statistics in /proc/meminfo.

- **total**—The amount of physical RAM on the platform.
- **free**—The amount of unused or available memory.
- **used**—The amount of allocated (permanent) and cached (temporary) memory.

The cache and buffers are not relevant to customer monitoring.

This information provides a general representation of the platform utilization only. You need more information to troubleshoot why memory utilization is high.

The **show processes memory** command displays the memory allocation per process.

```
switch# show processes memory
Load average: 1 minute: 0.43 5 minutes: 0.30 15 minutes: 0.28
Processes : 884 total, 1 running
CPU states : 2.0% user, 1.5% kernel, 96.5% idle
PID  MemAlloc MemLimit  MemUsed   StackBase/Ptr     Process
---- -------- --------- --------- ----------------- ----------------
4662 52756480 562929945 150167552 bfffdf00/bfffd970 netstack
```

Detailed Assessment of Platform Memory Utilization

Use the **show system internal memory-alerts-log** or the **show system internal kernel** command for a more detailed representation of memory utilization in Cisco NX-OS.

```
switch# show system internal kernel meminfo
MemTotal: 4135780 kB
MemFree: 578032 kB
Buffers: 5312 kB
Cached: 1926296 kB
RAMCached: 1803020 kB
Allowed: 1033945 Pages
Free: 144508 Pages
Available: 177993 Pages
SwapCached: 0 kB
Active: 1739400 kB
Inactive: 1637756 kB
HighTotal: 3287760 kB
HighFree: 640 kB
LowTotal: 848020 kB
LowFree: 577392 kB
SwapTotal: 0 kB
SwapFree: 0 kB
Dirty: 0 kB
Writeback: 0 kB
Mapped: 1903768 kB
Slab: 85392 kB
CommitLimit: 2067888 kB
Committed_AS: 3479912 kB
PageTables: 20860 kB
VmallocTotal: 131064 kB
VmallocUsed: 128216 kB
VmallocChunk: 2772 kB
```

In the output above, the most important fields are as follows:

- **MemTotal (kB)**—Total amount of memory in the system.
- **Cached (kB)**—Amount of memory used by the page cache (includes files in tmpfs mounts and data cached from persistent storage /bootflash).
- **RamCached (kB)**—Amount of memory used by the page cache that cannot be released (data not backed by persistent storage).
- **Available (Pages)**—Amount of free memory in pages (includes the space that could be made available in the page cache and free lists).
- **Mapped (Pages)**—Memory mapped into page tables (data being used by nonkernel processes).
- **Slab (Pages)**—Rough indication of kernel memory consumption.

Note One page of memory is equivalent to 4 kB of memory.

The **show system internal kernel memory global** command displays the memory usage for the page cache and kernel/process memory.

```
switch# show system internal kernel memory global
Total memory in system : 4129600KB
Total Free memory : 1345232KB
Total memory in use : 2784368KB
Kernel/App memory : 1759856KB
RAM FS memory : 1018616KB
```

Note In Cisco NX-OS, the Linux kernel monitors the percentage of memory that is used (relative to the total RAM present) and platform manager generates alerts as utilization passes default or configured thresholds. If an alert has occurred, it is useful to review the logs captured by the platform manager against the current utilization.

By reviewing the output of these commands, you can determine if the utilization is high as a result of the page cache, processes holding memory, or kernel.

Page Cache

If Cached or RAMCached is high, you should check the file system utilization and determine what kind of files are filling the page cache.

The **show system internal flash** command displays the file system utilization (the output is similar to df -hT included in the memory alerts log).

```
switch# show system internal flash
Mount-on                  1K-blocks      Used   Available   Use%  Filesystem
/                            409600     43008      367616     11   /dev/root
/proc                             0         0           0      0   proc
/sys                              0         0           0      0   none
/isan                        409600    269312      140288     66   none
/var/tmp                     307200       876      306324      1   none
/var/sysmgr                 1048576    999424       49152      96   none
/var/sysmgr/ftp              307200     24576      282624      8   none
/dev/shm                    1048576    412672      635904     40   none
/volatile                    204800         0      204800      0   none
/debug                         2048        16        2032      1   none
/dev/mqueue                       0         0           0      0   none
/mnt/cfg/0                    76099      5674       66496      8   /dev/hda5
/mnt/cfg/1                    75605      5674       66027      8   /dev/hda6
/bootflash                  1796768    629784     1075712     37   /dev/hda3
/var/sysmgr/startup-cfg      409600     27536      382064      7   none
/mnt/plog                     56192      3064       53128      6   /dev/mtdblock2
/dev/pts                          0         0           0      0   devpts
/mnt/pss                      38554      6682       29882     19   /dev/hda4
/slot0                      2026608         4     2026604      1   /dev/hdc1
/logflash                   7997912    219408     7372232      3   /dev/hde1
/bootflash_sup-remote      1767480   1121784      555912    67   127.1.1.6:/mnt/bootflash/
/logflash_sup-remote       7953616    554976     6994608      8   127.1.1.6:/mnt/logflash/
```

Note When reviewing this output, the value of none in the Filesystem column means that it is a tmpfs type.

In this example, utilization is high because the /var/sysmgr (or subfolders) is using a lot of space. /var/sysmgr is a tmpfs mount, which means that the files exist in RAM only. You need to determine what type of files are filling the partition and where they came from (cores/debugs/etc). Deleting the files will reduce utilization, but you should try to determine what type of files are taking up the space and what process left them in tmpfs.

Use the following commands to display and delete the problem files from the CLI:

- The **show system internal dir** *full directory path* command lists all the files and sizes for the specified path (hidden command).
- The **filesys delete** *full file path* command deletes a specific file (hidden command).

Kernel

Kernel issues are less common, but you can determine the problem by reviewing the slab utilization in the **show system internal kernel meminfo** command output. Generally, kernel troubleshooting requires Cisco customer support assistance to isolate why the utilization is increasing.

If slab memory usage grows over time, use the following commands to gather more information:

- The **show system internal kernel malloc-stats** command displays all the currently loaded KLMs, malloc, and free counts.

```
switch# show system internal kernel malloc-stats
Kernel Module Memory Tracking
-------------------------------------------------------------
Module           kmalloc  kcalloc  kfree    diff
klm_usd          00318846 00000000 00318825 00000021
klm_eobcmon      08366981 00000000 08366981 00000000
klm_utaker       00001306 00000000 00001306 00000000
klm_sysmgr-hb    00000054 00000000 00000049 00000005
klm_idehs        00000001 00000000 00000000 00000001
klm_sup_ctrl_mc  00209580 00000000 00209580 00000000
klm_sup_config   00000003 00000000 00000000 00000003
klm_mts          03357731 00000000 03344979 00012752
klm_kadb         00000368 00000000 00000099 00000269
klm_aipc         00850300 00000000 00850272 00000028
klm_pss          04091048 00000000 04041260 00049788
klm_rwsem        00000001 00000000 00000000 00000001
klm_vdc          00000126 00000000 00000000 00000126
klm_modlock      00000016 00000000 00000016 00000000
klm_e1000        00000024 00000000 00000006 00000018
klm_dc_sprom     00000123 00000000 00000123 00000000
klm_sdwrap       00000024 00000000 00000000 00000024
klm_obfl         00000050 00000000 00000047 00000003
```

By comparing several iterations of this command, you can determine if some KLMs are allocating a lot of memory but are not freeing/returning the memory back (the differential value will be very large compared to normal).

- The **show system internal kernel skb-stats** command displays the consumption of SKBs (buffers used by KLMs to send and receive packets).

```
switch# show system internal kernel skb-stats
Kernel Module skbuff Tracking
-------------------------------------------------------------
Module      alloc    free     diff
klm_shreth  00028632 00028625 00000007
klm_eobcmon 02798915 02798829 00000086
klm_mts     00420053 00420047 00000006
klm_aipc    00373467 00373450 00000017
klm_e1000   16055660 16051210 00004450
```

Compare the output of several iterations of this command to see if the differential value is growing or very high.

- The **show hardware internal proc-info slabinfo** command dumps all of the slab information (memory structure used for kernel management). The output can be large.

User Processes

If page cache and kernel issues have been ruled out, utilization might be high as a result of some user processes taking up too much memory or a high number of running processes (due to the number of features enabled).

Note Cisco NX-OS defines memory limits for most processes (rlimit). If this rlimit is exceeded, sysmgr will crash the process, and a core file is usually generated. Processes close to their rlimit may not have a large impact on platform utilization but could become an issue if a crash occurs.

Determining Which Process Is Using a Lot of Memory

The following commands can help you identify if a specific process is using a lot of memory:

- The **show process memory** command displays the memory allocation per process.

```
switch# show processes memory
PID   MemAlloc MemLimit   MemUsed    StackBase/Ptr     Process
----- -------- ---------- ---------- ----------------- ---------
4662  52756480 562929945  150167552  bfffdf00/bfffd970 netstack
```

Note The output of the **show process memory** command might not provide a completely accurate picture of the current utilization (allocated does not mean in use). This command is useful for determining if a process is approaching its limit.

- The **show system internal processes memory** command displays the process information in the memory alerts log (if the event occurred).

  To determine how much memory the processes are really using, check the Resident Set Size (RSS). This value will give you a rough indication of the amount of memory (in KB) that is being consumed by the processes. You can gather this information by using the **show system internal processes memory** command.

```
switch# show system internal processes memory
 PID TTY STAT TIME MAJFLT TRS RSS VSZ %MEM COMMAND
 4811 ?        Ssl  00:00:16      0    0 49772 361588  0.3 /isan/bin/routing-sw/clis
-cli /isan/etc/routing-sw/cli
 4928 ?       Ssl  00:18:41      0    0 44576 769512  0.2 /isan/bin/routing-sw/netstack
 /isan/etc/routing-sw/pm.cfg
 4897 ?        Ssl  00:00:18      0    0 42604 602216  0.2 /isan/bin/routing-sw/arp
 4791 ?        Ss   00:00:00      0    0 34384 318856  0.2 /isan/bin/pixm_vl
 4957 ?       Ssl  00:00:26      0    0 30440 592348  0.1 /isan/bin/snmpd -f -s udp:161
 udp6:161 tcp:161 tcp6:161
 5097 ?        Ssl  00:06:53      0    0 28052 941880  0.1 /isan/bin/routing-sw/pim -t

 5062 ?        Ss   00:01:00      0    0 27300 310596  0.1 /isan/bin/diag_port_lb
 5087 ?        Ssl  00:03:53      0    0 24988 992756  0.1 /isan/bin/routing-sw/bgp -t
 65001
 4792 ?        Ss   00:00:00      0    0 24080 309024  0.1 /isan/bin/pixm_gl
 5063 ?        Ss   00:00:01      0    0 21940 317440  0.1 /isan/bin/ethpm
 5044 ?        Ss   00:00:00      0    0 21700 304032  0.1 /isan/bin/eltm
 5049 ?        Ss   00:00:14      0    0 20592 306156  0.1 /isan/bin/ipqosmgr
 5042 ?        Ssl  00:00:05      0    0 20580 672640  0.1 /isan/bin/routing-sw/igmp
 5082 ?        Ssl  00:00:25      0    0 19948 914088  0.1 /isan/bin/routing-sw/mrib
-m 4
 5091 ?        Ssl  00:01:58      0    0 19192 729500  0.1 /isan/bin/routing-sw/ospfv3
 -t 8893
 5092 ?        Ssl  00:01:55      0    0 18988 861556  0.1 /isan/bin/routing-sw/ospf
-t 6464
 5083 ?        Ss   00:00:06      0    0 18876 309516  0.1 /isan/bin/mfdm
 remaining output omitted
```

If you see an increase in the utilization for a specific process over time, you should gather additional information about the process utilization.

Determining How a Specific Process Is Using Memory

If you have determined that a process is using more memory than expected, it is helpful to investigate how the memory is being used by the process.

- The **show system internal sysmgr service pid** *PID-in-decimal* command dumps the service information running the specified PID.

```
switch# show system internal sysmgr service pid 4727
Service "pixm" ("pixm", 109):
UUID = 0x133, PID = 4727, SAP = 176
State: SRV_STATE_HANDSHAKED (entered at time Fri May 10 01:42:01 2013).
Restart count: 1
Time of last restart: Fri May 10 01:41:11 2013.
The service never crashed since the last reboot.
Tag = N/A
Plugin ID: 1
```

Convert the UUID from the above output to decimal and use in the next command.

Note If you are troubleshooting in a lab, you can use Cisco NX-OS hexadecimal/decimal conversion using the following hidden commands:

- **hex**<decimal to convert>
- **dec**<hexadecimal to convert>

- The **show system internal kernel memory uuid** *uuid-in-decimal* command displays the detailed process memory usage including its libraries for a specific UUID in the system (convert UUID from the sysmgr service output).

```
switch# show system internal kernel memory uuid 307
 Note: output values in KiloBytes
 Name                          rss  shrd     drt  map    heap   ro  dat   bss stk misc

 ----                          ---  ----    ----  ---    ----   --  ---   --- --- ----

 /isan/bin/pixm                7816  5052    2764    1       0    0    0     0  52
0
 /isan/plugin/1/isan/bin/
pixm                         115472     0  115472    0  109176  752   28  6268   0
24
 /lib/ld-2.3.3.so                84    76       8    2       0   76    0     0   0
8
 /usr/lib/libz.so.1.2.1.1        16    12       4    1       0   12    4     0   0
0
 /usr/lib/libstdc++.so.6.0.3    296   272      24    1       0  272   20     4   0
0
 /lib/libgcc_s.so.1            1824    12    1812    1    1808   12    4     0   0
0
 /isan/plugin/1/isan/lib/
libtmifdb.so.0                   12     8       4    1       0    8    4     0   0
0
 /isan/plugin/0/isan/lib
libtmifdb_stub                   12     8       4    1       0    8    4     0   0
0
 /dev/mts                         0     0       0    0       1    0    0     0   0
0
 /isan/plugin/1/isan/lib/
libpcm_sdb.so.                   16    12       4    1       0   12    4     0   0
0
 /isan/plugin/1/isan/lib/
libethpm.so.0.                   76    60      16    1       0   60   16     0   0
0
 /isan/plugin/1/isan/lib
/libsviifdb.so.                  20     4      16    1      12    4    4     0   0
0
 /usr/lib/libcrypto.so.0.9.7    272   192      80    1       0  192   76     4   0
0
 /isan/plugin/0/isan/lib/
libeureka_hash                    8     4       4    1       0    4    4     0   0
0
 remaining output omitted
```

This output helps you to determine if a process is holding memory in a specific library and can assist with memory leak identification.

- The **show system internal** *service* **mem-stats detail** command displays the detailed memory utilization including the libraries for a specific service.

```
switch# show system internal pixm mem-stats detail
 Private Mem stats for UUID : Malloc track Library(103) Max types: 5
-----------------------------------------------------------------------------
TYPE NAME                                       ALLOCS                 BYTES
                                           CURR    MAX       CURR        MAX
   2 MT_MEM_mtrack_hdl                       35     35     132132     149940
   3 MT_MEM_mtrack_info                     598    866       9568      13856
   4 MT_MEM_mtrack_lib_name                 598    866      15860      22970
-----------------------------------------------------------------------------
Total bytes: 157560 (153k)
-----------------------------------------------------------------------------

Private Mem stats for UUID : Non mtrack users(0) Max types: 157
-----------------------------------------------------------------------------
TYPE NAME                                       ALLOCS                 BYTES
                                           CURR    MAX       CURR        MAX
   1 [0x41000000]ld-2.15.so                 283    283      48255      48256
   2 [0x41024000]libc-2.15.so               142    144       4979       5587
   8 [0x41241000]libglib-2.0.so.0.3200.3    500    771      10108      15588
  39 [0xf68af000]libindxobj.so                7      7        596        596
  45 [0xf68ca000]libavl.so                   73     73       1440       1440
  67 [0xf71b3000]libsdb.so                   56     58       3670      73278
  75 [0xf7313000]libmpmts.so                 35     37        280        380
  86 [0xf7441000]libutils.so                 23     28       3283       5766
  89 [0xf74bf000]libpss.so                   59     60       8564     483642
  90 [0xf750b000]libmts.so                    7      8        816        828
  92 [0xf754c000]libacfg.so                   0      4          0      51337
-----------------------------------------------------------------------------
Total bytes: 82817 (80k)
--------------------------------------------------------------------------------
 remaining output omitted
```

These outputs are usually requested by the Cisco customer support representative when investigating a potential memory leak in a process or its libraries.

Built-in Platform Memory Monitoring

Cisco NX-OS has built-in kernel monitoring of memory usage to help avoid system hangs, process crashes, and other undesirable behavior. The platform manager periodically checks the memory utilization (relative to the total RAM present) and automatically generates an alert event if the utilization passes the configured threshold values. When an alert level is reached, the kernel attempts to free memory by releasing pages that are no longer needed (for example, the page cache of persistent files that are no longer being accessed), or if critical levels are reached, the kernel will kill the highest utilization process. Other Cisco NX-OS components have introduced memory alert handling, such as the Border Gateway Protocol's (BGP's) graceful low memory handling, that allows processes to adjust their behavior to keep memory utilization under control.

Memory Thresholds

When many features are deployed, baseline memory requires the following thresholds:

- MINOR
- SEVERE

- CRITICAL

Because the default thresholds are calculated on boot up depending on the DRAM size, its value varies depending on the DRAM size that is used on the platform. The thresholds are configurable using the **system memory-thresholds minor** *percentage* **severe** *percentage* **critical** *percentage* command.

Beginning with Cisco NX-OS Release 10.2(4)M, the default system memory thresholds are as follows:

Beginning with Cisco NX-OS Release 10.3(1)F, the default system memory thresholds are as follows:

- Critical: 91
- Severe: 89
- Minor: 88

The **show system internal memory-status** command allows you to check the current memory alert status.

```
switch# show system internal memory-status
MemStatus: OK
```

Switches running scaled deployment, including scaled BGP EVPN VxLAN VNI (please see *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide* for supported scale), the memory alert may be seen during Non-Disruptive ISSU as the default system memory threshold has been lowered beginning with Cisco NX-OS Release 10.3(3)F release. To avoid system reacting to critical memory alert, before upgrade configure higher value for system memory thresholds. For example: Set system memory thresholds as 90 for minor, 94 for severe, and 95 for critical.

Memory Alerts

When a memory threshold has been passed (OK -> MINOR, MINOR -> SEVERE, SEVERE -> CRITICAL), the Cisco NX-OS platform manager captures a snapshot of memory utilization and logs an alert to syslog. This snapshot is useful in determining why memory utilization is high (process, page cache, or kernel). The log is generated in the Linux root path (/) and copy is moved to OBFL (/mnt/plog) if possible. This log is very useful for determining if memory utilization is high due to the memory that was consumed by the page cache, kernel, or Cisco NX-OS user processes.

The **show system internal memory-alerts-log** command displays the memory alerts log.

The memory alerts log consists of the following outputs:

Command	Description
cat /proc/memory_events	Provides a log of time stamps when memory alerts occurred.
cat /proc/meminfo	Shows the overall memory statistics including the total RAM, memory consumed by the page cache, slabs (kernel heap), mapped memory, available free memory, and so on.
cat /proc/memtrack	Displays the allocation/deallocation counts of the KLMs (Cisco NX-OS processes running in kernel memory).
df -hT	Displays file system utilization information (with type).
du --si -La /tmp	Displays file information for everything located in /tmp (symbolic link to /var/tmp).

Command	Description
cat /proc/memory_events	Dumped a second time to help determine if utilization changed during data gathering.
cat /proc/meminfo	Dumped a second time to help determine if utilization changed during data gathering.

LPSS Shared Memory Monitoring

Beginning with Cisco NX-OS 10.5(1)F, LPSS (Lightweight Persistent Storage Service) shared memory monitoring feature is introduced. Users can use this feature to monitor the usage of shared memory by LPSS. This feature is automatically enabled. This feature is supported on all Nexus 9000 and 3000 series switches.

Disabling LPSS Shared Memory Monitoring

SUMMARY STEPS

1. **configure**
2. (Optional)**system lpss monitor**
3. (Optional)**frequency** *frequency*
4. (Optional)**threshold** *threshold*
5. **no system lpss monitor**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**configure**
	Example:
	`switch# configure`
	`switch(config)#`
		Enters configuration mode.

Step 2	(Optional)**system lpss monitor** **Example:** `switch(config)# system lpss monitor`	Configures LPSS monitoring.
Step 3	(Optional)**frequency** *frequency* **Example:** `switch(config-lpss)# frequency 8`	Configures monitoring frequency. When the frequency is reached, syslog is generated. Frequency default – 10 (multiples of 500 milliseconds) Frequency range – 1-10

	Command or Action	Purpose
Step 4	(Optional)**threshold** *threshold* **Example:** `switch(config-lpss)# threshold 80`	Configures monitoring threshold. When the threshold is reached, syslog is generated. Threshold default – 100 (in percentage) Threshold range – 70-100 of shared memory
Step 5	**no system lpss monitor** **Example:** `switch(config-lpss)# system lpss monitor`	Disables this feature. This feature is automatically enabled.

Verifying LPSS Shared Memory Monitoring Configuration

Use the following command to see the LPSS usage details.

```
switch# show system lpss monitor usage
Total SHM size: 6400 MB
Total LPSS Shared memory usage: 756 MB (11%)
Monitoring Frequency: 8
Total Threshold: 80
switch#
```

CHAPTER 11

Troubleshooting Packet Flow Issues

Packet Flow Issues

Packets could be dropped for the following reasons:

- Software-switched packets could be dropped because of Control Plane Policing (CoPP).
- Hardware-switched packets could be dropped by the hardware because of a bandwidth limitation.

Beginning with Cisco NX-OS Release 10.3.(1)F, the following CLIs are supported on Cisco Nexus 9300 and 9500 Cloud Scale switches.

- **show hardware internal statistics module-all all**: Displays the statistics of active modules.
- **show hardware internal statistics module <module-no> all**: Displays the statistics of a particular active module from supervisor.

Packets Dropped Because of Rate Limits

Use the **show hardware rate-limit** command to determine if packets are being dropped because of a rate limit.

```
switch(config)# show hardware rate-limit module 1

Units for Config: packets per second
Allowed, Dropped & Total: aggregated since last clear counters

Rate Limiter Class                        Parameters
------------------------------------------------------------
access-list-log                          Config    : 100
                                         Allowed   : 0
                                         Dropped   : 0
                                         Total     : 0
```

Packets Dropped Because of CoPP

Use the **show policy-map interface control-plane** command to determine if packets are being dropped because of CoPP.

```
switch# show policy-map interface control-plane
   class-map copp-system-p-class-exception (match-any)
      match exception ip option
      match exception ip icmp unreachable
      match exception ttl-failure
      match exception ipv6 option
      match exception ipv6 icmp unreachable
      match exception mtu-failure
      set cos 1
      police cir 200 pps , bc 32 packets

      module 27 :
        transmitted 0 packets;
        dropped 0 packets;

      module 28 :
        transmitted 0 packets;
        dropped 0 packets;
```

Monitoring Inband Packet Statistics

Use the **show hardware internal cpu-mac inband counters** command to display inband packet statistics for supervisor modules, fabric modules, and line cards.

```
switch# show hardware internal cpu-mac inband counters
eth2 counters:
eth2      Link encap:Ethernet  HWaddr 00:00:00:01:1b:01
          BROADCAST MULTICAST  MTU:9400  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth3 counters:
eth3      Link encap:Ethernet  HWaddr 00:00:00:01:1b:01
          inet6 addr: fe80::200:ff:fe01:1b01/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9400  Metric:1
          RX packets:425432 errors:0 dropped:0 overruns:0 frame:0
          TX packets:352432 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:253284953 (241.5 MiB)  TX bytes:249647978 (238.0 MiB)

ps-inb counters:
ps-inb    Link encap:Ethernet  HWaddr 00:00:00:01:1b:01
          inet6 addr: fe80::200:ff:fe01:1b01/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9400  Metric:1
          RX packets:128986 errors:0 dropped:0 overruns:0 frame:0
          TX packets:129761 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:221538103 (211.2 MiB)  TX bytes:227158091 (216.6 MiB)


switch# slot 22 show hardware internal cpu-mac inband  counters
inband0 counters:
inband0   Link encap:Ethernet  HWaddr 00:00:00:01:16:03
```

```
          inet addr:127.2.2.22  Bcast:127.2.255.255  Mask:255.255.0.0
          inet6 addr: fe80::200:ff:fe01:1603/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9676  Metric:1
          RX packets:147425 errors:0 dropped:0 overruns:0 frame:0
          TX packets:147470 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:532
          RX bytes:15479625 (14.7 MiB)  TX bytes:14898335 (14.2 MiB)
          Interrupt:10

knet0_0 counters:
knet0_0   Link encap:Ethernet  HWaddr 02:10:18:e1:6f:50
          inet6 addr: fe80::10:18ff:fee1:6f50/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9400  Metric:1
          RX packets:36 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:6 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3816 (3.7 KiB)  TX bytes:0 (0.0 B)

knet0_1 counters:
knet0_1   Link encap:Ethernet  HWaddr 02:10:18:e1:6f:51
          inet6 addr: fe80::10:18ff:fee1:6f51/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9400  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:6 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Fabric Connectivity Commands

Cisco NX-OS provides the following commands to display information and statistics related to fabric connectivity:

- **show system internal fabric connectivity** [**module** *module-number*]—Displays connectivity information for all fabric modules or a single module.

```
switch# show system internal fabric connectivity
HiGIG Link-info Linecard slot:4
--------------------------------------------------------------
 LC-Slot  LC-Unit   LC-HGLink   FM-Slot   FM-Unit   FM-HGLink
--------------------------------------------------------------
    4        0       HG02         22        0        HG09
    4        0       HG03         22        1        HG09
    4        0       HG06         24        0        HG09
    4        0       HG07         24        1        HG09
    4        1       HG02         22        0        HG10
    4        1       HG03         22        1        HG10
    4        1       HG06         24        0        HG10
    4        1       HG07         24        1        HG10
    4        2       HG02         22        0        HG11
    4        2       HG03         22        1        HG11
    4        2       HG06         24        0        HG11
    4        2       HG07         24        1        HG11

HiGIG Link-info Fabriccard slot:22
--------------------------------------------------------------
 FM-Slot  FM-Unit   FM-HGLink   LC-Slot   LC-Unit   LC-HGLink
--------------------------------------------------------------
   22        0       HG09          4        0        HG02
   22        0       HG10          4        1        HG02
   22        0       HG11          4        2        HG02
   22        1       HG09          4        0        HG03
   22        1       HG10          4        1        HG03
   22        1       HG11          4        2        HG03

HiGIG Link-info Fabriccard slot:24
--------------------------------------------------------------
 FM-Slot  FM-Unit   FM-HGLink   LC-Slot   LC-Unit   LC-HGLink
--------------------------------------------------------------
   24        0       HG09          4        0        HG06
   24        0       HG10          4        1        HG06
   24        0       HG11          4        2        HG06
   24        1       HG09          4        0        HG07
   24        1       HG10          4        1        HG07
   24        1       HG11          4        2        HG07
```

- **show system internal interface counters module** *module-number* [**nz**]—Displays the rates for HG or fabric links on a module. The **nz** option displays only non-zero counters.

```
switch# show system internal interface counters module 22 nz
Internal Port Counters (150 secs rate) for Slot: 22
====================================================

Interface ASIC ASIC  BCM  TxBitRate(BwUtil) TxPktRate RxBitRate(BwUtil) RxPktRate
          Port Inst Port     (bps)             (pps)       (bps)          (pps)
----------------------------------------------------------------------------------
ii22/1/10  HG9    0   10     0(  0.00)         0       33064(  0.00)      17

switch# show system internal interface counters module 22
Internal Port Counters (150 secs rate) for Slot: 22
====================================================

Interface ASIC ASIC  BCM  TxBitRate(BwUtil) TxPktRate RxBitRate(BwUtil) RxPktRate
          Port Inst Port      (bps)            (pps)       (bps)           (pps)
----------------------------------------------------------------------------------
ii22/1/1  HG0    0    1       0(  0.00)        0         0(  0.00)         0
ii22/1/2  HG1    0    2       0(  0.00)        0         0(  0.00)         0
ii22/1/3  HG2    0    3       0(  0.00)        0         0(  0.00)         0
ii22/1/4  HG3    0    4       0(  0.00)        0         0(  0.00)         0
ii22/1/5  HG4    0    5       0(  0.00)        0         0(  0.00)         0
ii22/1/6  HG5    0    6       0(  0.00)        0         0(  0.00)         0
ii22/1/7  HG6    0    7       0(  0.00)        0         0(  0.00)         0
ii22/1/8  HG7    0    8       0(  0.00)        0         0(  0.00)         0
ii22/1/9  HG8    0    9       0(  0.00)        0         0(  0.00)         0
ii22/1/10  HG9    0   10      0(  0.00)        0     30888(  0.00)        12
ii22/1/11 HG10    0   11      0(  0.00)        0         0(  0.00)         0
ii22/1/12 HG11    0   12      0(  0.00)        0         0(  0.00)         0
ii22/1/13 HG12    0   13      0(  0.00)        0         0(  0.00)         0
ii22/1/14 HG13    0   14      0(  0.00)        0         0(  0.00)         0
ii22/1/15 HG14    0   15      0(  0.00)        0         0(  0.00)         0
ii22/1/16 HG15    0   16      0(  0.00)        0         0(  0.00)         0
ii22/1/17 HG16    0   17      0(  0.00)        0         0(  0.00)         0
ii22/1/18 HG17    0   18      0(  0.00)        0         0(  0.00)         0
ii22/1/19 HG18    0   19      0(  0.00)        0         0(  0.00)         0
ii22/1/20 HG19    0   20      0(  0.00)        0         0(  0.00)         0
ii22/1/21 HG20    0   21      0(  0.00)        0         0(  0.00)         0
ii22/1/22 HG21    0   22      0(  0.00)        0         0(  0.00)         0
ii22/1/23 HG22    0   23      0(  0.00)        0         0(  0.00)         0
ii22/1/24 HG23    0   24      0(  0.00)        0         0(  0.00)         0
ii22/1/33  HG0    1    1      0(  0.00)        0         0(  0.00)         0
ii22/1/34  HG1    1    2      0(  0.00)        0         0(  0.00)         0
ii22/1/35  HG2    1    3      0(  0.00)        0         0(  0.00)         0
ii22/1/36  HG3    1    4      0(  0.00)        0         0(  0.00)         0
ii22/1/37  HG4    1    5      0(  0.00)        0         0(  0.00)         0
```

```
ii22/1/38  HG5    1    6      0(  0.00)        0         0(  0.00)          0
ii22/1/39  HG6    1    7      0(  0.00)        0         0(  0.00)          0
ii22/1/40  HG7    1    8      0(  0.00)        0         0(  0.00)          0
ii22/1/41  HG8    1    9      0(  0.00)        0         0(  0.00)          0
ii22/1/42  HG9    1   10      0(  0.00)        0         0(  0.00)          0
ii22/1/43 HG10    1   11      0(  0.00)        0         0(  0.00)          0
ii22/1/44 HG11    1   12      0(  0.00)        0         0(  0.00)          0
ii22/1/45 HG12    1   13      0(  0.00)        0         0(  0.00)          0
ii22/1/46 HG13    1   14      0(  0.00)        0         0(  0.00)          0
ii22/1/47 HG14    1   15      0(  0.00)        0         0(  0.00)          0
ii22/1/48 HG15    1   16      0(  0.00)        0         0(  0.00)          0
ii22/1/49 HG16    1   17      0(  0.00)        0         0(  0.00)          0
ii22/1/50 HG17    1   18      0(  0.00)        0         0(  0.00)          0
ii22/1/51 HG18    1   19      0(  0.00)        0         0(  0.00)          0
ii22/1/52 HG19    1   20      0(  0.00)        0         0(  0.00)          0
ii22/1/53 HG20    1   21      0(  0.00)        0         0(  0.00)          0
ii22/1/54 HG21    1   22      0(  0.00)        0         0(  0.00)          0
ii22/1/55 HG22    1   23      0(  0.00)        0         0(  0.00)          0
ii22/1/56 HG23    1   24      0(  0.00)        0         0(  0.00)          0
```

- **show system internal interface counters detail module** *module-number*—Displays detailed statistics for all HG or fabric links on a single module.

```
show system internal interface counters detail module 4
................
............
Interface: ii4/1/3  ASIC Inst# 0/Port# 3/Name HG2
--------------------------------------------------------
Last Cleared @ Thu Jan  1 00:00:00 2013
 (0)
Tx/Rx Rates (per second):
          secs     tx bytes    tx packets  rx bytes    rx packets
    [0] - 10       0           0           0           0
    [1] - 150      9448        60          0           0
    [2] - 300      9448        60          0           0
Mac Pktflow:
  Rx Counters:
    Ingress Packets  :       0x0000000000000000/0
    Unicast Packets  :       0x0000000000000000/0
    Multicast Packets:       0x0000000000000000/0
    Broadcast Packets:       0x0000000000000000/0
    Jumbo Packets    :       0x0000000000000000/0
    Total Bytes      :       0x0000000000000000/0

    Rx Bytes by Packet Size:
    64:             0x0000000000000000/0
    65   - 127:     0x0000000000000000/0
    128  - 255:     0x0000000000000000/0
    256  - 511:     0x0000000000000000/0
    512  - 1023:    0x0000000000000000/0
    1024 - 1518:    0x0000000000000000/0
    1519 - 1548:    0x0000000000000000/0

  Tx Counters:
    Egress Packets   :       0x0000000000001351/4945
    Unicast qackets:         0x0000000000001351/4945
    Multicast qackets:       0x0000000000000000/0
    Broadcast Packets:       0x0000000000000000/0
    Jumbo Packets    :       0x0000000000000000/0
    Undersize Packets:       0x0000000000000000/0
    Total Bytes      :       0x000000000008e756/583510

    Tx Bytes by Packet Size
    64:             0x0000000000000000/0
    65   - 127:     0x0000000000001351/4945
    128  - 255:     0x0000000000000000/0
    256  - 511:     0x0000000000000000/0
    512  - 1023:    0x0000000000000000/0
    1024 - 1518:    0x0000000000000000/0
    1519 - 1548:    0x0000000000000000/0
    trunk:          0x0000000000000000/0

Mac Control:
    Rx Pause:       0x0000000000000000/0
    Tx Pause:       0x0000000000000000/0
    Reset:          0x0000000000000000/0
Mac Errors:
    Undersize:      0x0000000000000000/0
    Runt:           0x0000000000000000/0
    Crc:            0x0000000000000000/0
    Input Errors:   0x0000000000000000/0
    In Discard:     0x0000000000000000/0
    Giants:         0x0000000000000000/0
    Output Errors:  0x0000000000000000/0
    Output Discard: 0x0000000000000000/0
    Bad Proto:      0x0000000000000000/0
    Collision:      0x0000000000000000/0
    Late Collision: 0x0000000000000000/0
    No Carrier:     0x0000000000000000/0
```

Troubleshooting Packet Flow with Packet Tracer

Packet Tracer

The Packet Tracer is a new troubleshooting tool that allows a packet to be captured from the Network Processor. Similar to the ELAM tool available on Cisco Nexus 9000 Cloud Scale switches, this tool provides information to understand how the ASIC forwarded the captured packet. This information is useful to troubleshoot packet flow.

While various tools like SPAN, ERSPAN, Ethanalyzer exist to debug packet flow issues, Packet Tracer allows troubleshooting within the forwarding pipeline of an ASIC without any performance penalty or disruption to the environment.

Packet tracer has the following capabilities:

Important To effectively use Packet Tracer, users should have a comprehensive understanding of the ASIC forwarding pipeline. This knowledge is crucial for setting precise filters and accurately interpreting the results of packet captures.

- Allows filters for capturing packets based on various protocol parameters like IPv4/IPv6 addresses, TCP/UDP ports, and so on.

- Provides flexible filtering functionality at the ASIC level due to matching incoming packets using a 128-byte filter and 128-byte mask. This allows the filters to be designed using packet protocol parameters or specific byte sequences in the payload portion of the packet.

- Captures 128 bytes of packet along with the state of the forwarding pipeline because the packet passes through various stages of the pipeline.
- Allows creation of filters using non-protocol information like packet size, traffic class, and so on.

Packet Tracer Workflow

You must perform the following steps to capture packets using Packet Tracer.

- Determine whether the packet must be captured in the receive packet path (RxPP) or transmit packet path (TxPP).
- Determine whether a packet must be captured using protocol parameters or specific byte sequences as filters.
  - To capture packets using protocol parameters as filters: Identify the frame or packet format of the packet to be captured which will guide you to know whether the packet to be captured is an Ethernet packet followed by IPv4 or IPv6 or other well-known protocols or a VLAN tagged Ethernet packet followed by well-known protocols and so on. For more information, see Packet Format, on page 103.

Note The packet format serves as a template for forming the 128-byte filter and mask based on various protocol parameters that you may select as filters.

  - To capture packets using specific byte sequences as filters: Identify the specific contiguous byte sequences and start offset.
- Start the Packet Tracer.
- Wait for the Packet Tracer to trigger and display the result.

Packet Format

The Packet Tracer on ASIC uses a 128-byte packet filter and a corresponding 128-byte filter mask to capture specific packets. You use the packet filter as a byte string representing the first 128 bytes of the packet of interest, with specific values at certain offsets. The ASIC matches ingress/egress packets against this byte pattern using the filter mask. This setup allows you to capture packets of any protocol type or pattern, providing flexibility in specifying the match pattern.

To build the packet filter and mask, you must typically identify protocol fields of interest, like dot1q header, IPv4 source IP, destination IP, or UDP source port, and incorporate these values at the correct offsets in the packet filter. The matching mask is then applied at the same offsets in the filter mask. Because packets are matched against the 128-byte filter and mask, you must be aware of the specific packet format to create a representative pattern including ethernet headers.

This image represents the packet format tree that can help create specific packet format.

Figure 1: Packet Format Tree

```
eth
+-- arp
+-- ipv4
¦ +-- tcp
¦ +-- udp
¦ ¦ +-- vxlan
¦ ¦ +-- eth
¦ ¦ ¦ +-- arp
¦ ¦ ¦ +-- llcu
¦ ¦ ¦ ¦ +-- snap
¦ ¦ ¦ +-- llci
¦ ¦ ¦ ¦ +-- snap
¦ ¦ ¦ +-- llcs
¦ ¦ ¦ ¦ +-- snap
¦ ¦ ¦ +-- ipv4
¦ ¦ ¦ ¦ +-- tcp
¦ ¦ ¦ ¦ +-- udp
¦ ¦ ¦ ¦ +-- icmp
¦ ¦ ¦ +-- ipv6
¦ ¦ ¦ ¦ +-- tcp
¦ ¦ ¦ ¦ +-- udp
¦ ¦ ¦ ¦ +-- icmpv6
¦ ¦ ¦ +-- qinq
¦ ¦ ¦ ¦ +-- ipv4
¦ ¦ ¦ ¦ ¦ +-- tcp
¦ ¦ ¦ ¦ ¦ +-- udp
¦ ¦ ¦ ¦ ¦ +-- icmp
¦ ¦ ¦ ¦ +-- ipv6
¦ ¦ ¦ ¦ ¦ +-- tcp
¦ ¦ ¦ ¦ ¦ +-- udp
¦ ¦ ¦ ¦ ¦ +-- icmpv6
¦ ¦ ¦ +-- vntag
¦ ¦ ¦ ¦ +-- ipv4
¦ ¦ ¦ ¦ ¦ +-- tcp
¦ ¦ ¦ ¦ ¦ +-- udp
¦ ¦ ¦ ¦ ¦ +-- icmp
¦ ¦ ¦ ¦ +-- ipv6
¦ ¦ ¦ ¦ ¦ +-- tcp
¦ ¦ ¦ ¦ ¦ +-- udp
¦ ¦ ¦ ¦ ¦ +-- icmpv6
¦ ¦ +-- mpls
¦ ¦ ¦ +-- mpls
¦ ¦ ¦ ¦ +-- ipv4
¦ ¦ ¦ ¦ +-- ipv6
¦ ¦ +-- arp
¦ +-- icmp
¦ +-- gre
¦ ¦ +-- ipv4
¦ ¦ +-- ipv6
¦ +-- dot1q
```

This table lists a few examples of supported packet format usage of packet tracer.

Table 3: Examples of Supported Packet Format

Interested traffic	Packet-format to use
TCP traffic with VLAN tagging	• For IPv4: eth-dot1q-ipv4-tcp • For IPv6: eth-dot1q-ipv6-tcp

Interested traffic	Packet-format to use
VXLAN traffic with VLAN tagging	• For IPv4: eth-dot1q-ipv4-udp-vxlan-eth-ipv4 • For IPv6: eth-dot1q-ipv6-udp-vxlan-eth-ipv4
VXLAN traffic with inner VLAN tagging	• ForIPv4:eth-ipv4-udp-vxlan-eth-dot1q-ipv4 • For IPv6:eth-ipv6-udp-vxlan-eth-dot1q-ipv4
VXLAN traffic with inner VLAN tagging and ARP	• For IPv4: eth-ipv4-udp-vxlan-eth-dot1q-arp • For IPv6: eth-ipv6-udp-vxlan-eth-dot1q-arp
ARP traffic	eth-arp
ICMP traffic	• For IPv4: eth-ipv4-icmp • For IPv6: eth-ipv6-icmp
MPLS traffic	• eth-mpls-mpls-ipv4 (MPLS can be added up to 6 times for 6 labels)

Guidelines and Limitations for Packet Tracer

The Packet Tracer has the following configuration guidelines and limitations:

- In the packet offset command, a maximum of 10 conditions can be set for RxPP or TxPP.
- Packet filter matches only on the first 128 Bytes of the packet.
- Packet tracer can be done in either RxPP or TxPP path.
- Capture mode ‘continuous’ is not supported.
- If slices are not specified, the output is displayed for all the slices where the packet-trace is tracked.

Supported Release and Platform for Packet Tracer

Release	Platform
10.5(3)F and later	Cisco N9364E-SG2-Q and N9364E-SG2-O switches
10.6(1)F and later	Cisco N9336C-SE1 switches

Deploy Packet Tracer

You can use the following commands to trigger packet tracer functionality on the switch.

Procedure

Step 1 Run the **packet-trace** command to enable the packet trace mode.

Example:

```
switch# packet-trace
switch(config-pt)#
```

Step 2 Run the **trigger init** {**rxpp** | **txpp**} command to capture the packet on Rx path (Rxpp) or Tx Path (TxPP).

Example:

```
switch(config-pt)# trigger init rxpp
switch(config-pt-rxpp)#
```

Step 3 Run the **packet-format** *packet-format* command to specify the packet format to trace the packets.

Example:

```
switch(config-pt-txpp)# packet-format eth-ipv4-tcp
switch(config-pt-txpp-pkt-fmt)#
```

In this example, the frame format of packet to be captured is Ethernet with IPv4 and TCP protocol filter.

Note
This command provides a User Guide option under help (?) to guide on the usage of this command.

Step 4 Run the **set outer** {**l2** | **ipv4** | **ipv6** | **arp** | **l4** | **mpls** } | **set inner** { **l2** | **ipv4** | **ipv6** | **arp** | **l4** | **mpls** } command to set various protocol filters for nonencapsulated packets (set outer) or encapsulated packet (set inner).

Example:

```
switch(config-pt-txpp-pkt-fmt)# set outer ipv4
```

To set filters for packet, an appropriate packet-format must be selected, see Packet Format, on page 103 section.

Note
For encapsulated packet, the term "outer" is used for underlay protocols like IP/UDP and the term "inner" is used for overlay protocols like VXLAN, GRE, and so on.

Step 5 (Optional) Run the **show filters** command to set various nonpacket filters.

Example:

```
switch(config-pt-txpp-pkt-fmt)# show filters
```

These filters can be set in addition to packet header filters.

Note
Packet capture is based on both packet header filters and nonpacket filters, if set.

Step 6 Run the **start** command to set the filters on the ASIC and begins the capture operation.

Example:

```
switch(config-pt-txpp-pkt-fmt)# start
```

Step 7 (Optional) Run the **status** command to provide information on whether capture has occurred or not.

Example:

```
switch(config-pt-txpp-pkt-fmt)# status
```

This command can be run multiple times to know whether a packet has been captured or not.

Step 8 (Optional) Run the **stop** command to cancel the capture that has been started.

Example:

```
switch(config-pt-txpp-pkt-fmt)# stop
```

Note
Issuing a **stop** command after a successful capture does not alter the capture's outcome.

Step 9 (Optional) Run the **reset** command to clear the set filters.

Example:

```
switch(config-pt-txpp-pkt-fmt)# reset
switch(config-pt-txpp)#
```

Step 10 Run the **report** [**brief**] [**slice** *slice-ids*] command to provide details of the captured packet and dump of NPPD decode information for each stage of the capture.

Example:

```
switch(config-pt-txpp)# report detail
```

For **brief** option, NPPD decoded information will not be dumped.

If a **slice** option is specified, packet capture details of the specified slice will be displayed.

For more details of the report detail, see Configuration Example for Packet Tracer, on page 109.

Step 11 Run the **set npi** {**err-flag** *value* | **initial-TC** *value* | **processing-code** *value* | **reassembly-ctxt** *value* | **single-frag-pkt** *value* | **src-pif** *value* | **tx-to-rx-rec-data** *value* | **unsch-rec-code** *value*} command to set various nonpackets in RxPP capture mode.

Example:

```
switch(config-pt-rxpp)# set npi src-pif 15
```

These filters can be set in addition to packet header filters.

Packet capture is based on both packet header filters and nonpacket filters, if set.

- **Error Flag (err-flag)**: Use this option to check if any hardware errors are observed. Size: 1 bit. Range: 0–1.
- **Initial TC (initial-TC)**: Use this option to set the packets initial traffic class as calculated by IFG. Size: 3 bits. Range: 0–7.
- **Processing Code (processing-code)**: Use this option to indicate abnormal events. Size: 7 bits. Range: 0–127.
- **Reassembly Context (reassembly-ctxt)**: Use this option to uniquely identify the packet fragments for reassembly. Size: 11 bits. Range: 0–2047.
- **Single Fragment Packet (single-frag-pkt)**: Use this option to set a flag for single fragment packet. Size: 1 bit. Range: 0–1.
- **Source Port interface (src-pif)**: Use this option to set the source physical interface. Size: 7 bits. Range: 0–127.
- **Tx to Rx recycle data (tx-to-rx-rec-data)**: Use this option to set recycle data passed from the TxNPU to the RxNPU. Size: 8 bits. Range: 0–255.

- **Unschedule Recycle code (unsch-rec-code)**: Use this option to set the type of unscheduled recycle required. Size: 4 bits. Range: 0–15.

Step 12 Run the **set npi** {**acc-LM-cache-and-Idx** *value* | **colour** *value* | **congested** *value* | **congestion-level** *value* | **cud** *value* | **dst-intf** *value* | **eop** *value* | **err-flag** *value* | **is-elephant-flow** *value* | **lm-cache-index** *value* | **omd** *value* | **pkt-size** *value* | **sop** *value* | **src-slice** *value* | **src-slice-sys-port** *value* | **start-packing** *value* | **traffic-class** *value*} command to set various nonpackets in TxPP capture mode.

Example:

```
switch(config-pt-txpp)# set npi dst-intf 25
```

These filters can be set in addition to packet header filters.

Packet capture is based on both packet header filters and nonpacket filters, if set.

- **Access LM cache and Index (acc-LM-cache-and-Idx)**: Use this option to check the loss measurement request cache. Size: 1 bit. Range: 0–1.
- **Color (colour)**: Use this option to check drop precedence of the packet. Size: 2 bits. Range: 0–3.
- **Congested (congested)**: Use this option to check whether a congestion event occurred on the packet. Size: 1 bit. Range: 0–1.
- **Congestion Level (congestion-level)**: Use this option to check measurement of the queues congestion as experienced by the packet. Size: 4 bits. Range: 0–15.
- **Copy unique data (cud)**: Use this option to check why the packet copy was generated. Size: 23 bits. Range: 0–8388607
- **Destination Interface (dst-intf)**: Use this option to set the destination physical interface. Size: 8 bits. Range: 0–255.
- **End Of Packet (eop)**: Use this option to set an end of a packet fragment. Size: 1 bit. Range: 0–1.
- **Error Flag (err-flag)**: Use this option to check if any hardware errors are observed. Size: 1 bit. Range: 0–1.
- **Is elephant flow (is-elephant-flow)**: Use this option to check if a packet is identified as part of a large flow. Size: 1 bit. Range: 0–1.
- **LM Cache Index (lm-cache-index)**: Use this option to check index of the loss measurement request cache. Size: 2 bits. Range: 0–255.
- **Output queue Mapped Data (omd)**: Use this option to set output queue-mapped data. Size: 9 bits. Range: 0–511.
- **Packet size (pkt-size)**: Use this option to check packet size in bytes. Size: 14 bits.
- **Start Of Packet (sop)**: Use this option to set the start of a packet fragment. Size: 1 bit. Range: 0–1.
- **Source Slice (src-slice)**: Use this option to set source slice. Size: 3 bits.
- **Source Slice System Port (src-slice-sys-port)**: Use this option to set the source slice system port. Size: 8 bits. Range: 0–255.
- **Start Packing (start-packing)**: Use this option to set the first packet of a dual packet. Size: 1 bit. Range: 0–1.
- **Traffic Class (traffic-class)**: Use this option to set the traffic class of the packet. Size: 3 bits. Range: 0–7.

Step 13 Run the **set pkt-offset condition** *Value* **offset** *Value* command to set the raw filter and mask to any byte pattern according to your needs.

Example:

```
switch(config-pt-txpp)# set pkt-offset condition 1 offset 0x10 value 0xababab mask 0xffffff
```

Up to 10 such conditions can be set. This operation is mutually exclusive to set the protocol filters using the **set outer** command.

Note

- A mask of '0xf' indicates that the corresponding nibble in the filter should be considered, whereas a mask of '0x0' means it should be ignored.
- This command must be used only if you know the offset value.

Verification of the Packer Tracer Deployment

Use the following command to display Packer Tracer deployment information.

Command	Purpose
show filters	Displays the actual pkt-offset filters set. For more information, see Configuration Example for Packet Tracer, on page 109.

Configuration Example for Packet Tracer

The following examples shows how Packet Tracer functionality is used to capture filters and reports:

```
switch# packet-trace

switch(config-pt)# trigger init rxpp

switch(config-pt-rxpp)# packet-format eth-dot1q-ipv4

switch(config-pt-rxpp-pkt-fmt)#
switch(config-pt-rxpp-pkt-fmt)# set outer ipv4 src-ip 62.0.134.2

switch(config-pt-rxpp-pkt-fmt)# set outer ipv
ipv4   ipv6
switch(config-pt-rxpp-pkt-fmt)# set outer ipv4 next-protocol 17

switch(config-pt-rxpp-pkt-fmt)#
switch(config-pt-rxpp-pkt-fmt)# show filters

slot  1
=======

OUTER
    Ethernet
        eth_type:: value: 0x800, mask: 0xffff, offset: 16
    DOT1Q
        tpid:: value: 0x8100, mask: 0xffff, offset: 12
    IPv4
        protocol:: value: 17, mask: 0xff, offset: 27
        src_ip:: value: 62.0.134.2, mask: 0xffffffff, offset: 30

INNER
```

```
Packet filters::
0000000000000000000000008100000008000000000000000000001100003e00860200000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
Packet filters mask::
000000000000000000000000ffff0000ffff000000000000000000ff0000ffffffff00000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000


switch(config-pt-rxpp-pkt-fmt)#
switch(config-pt-rxpp-pkt-fmt)#
switch(config-pt-rxpp-pkt-fmt)# start
Filter name: eth_type
Filter name: tpid
Filter name: protocol
Filter name: src_ip
Setting RX packet data filters (filter count 4).

switch(config-pt-rxpp-pkt-fmt)#
switch(config-pt-rxpp-pkt-fmt)#
switch(config-pt-rxpp-pkt-fmt)#
switch(config-pt-rxpp-pkt-fmt)#
switch(config-pt-rxpp-pkt-fmt)# status
Packet trace hit

switch(config-pt-rxpp-pkt-fmt)#
switch(config-pt-rxpp-pkt-fmt)#
switch(config-pt-rxpp-pkt-fmt)# report
Rx Packet traced in slot #0 slice #1:

Frame 1: 128 bytes on wire (1024 bits), 128 bytes captured (1024 bits)
    [Protocols in frame: eth:ethertype:vlan:ethertype:ip:udp:hsrp]
Ethernet II, Src: Cisco_21:e5:5b (40:14:82:21:e5:5b), Dst: IPv4mcast_66 (01:00:5e:00:00:66)

    Destination: IPv4mcast_66 (01:00:5e:00:00:66)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    Source: Cisco_21:e5:5b (40:14:82:21:e5:5b)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: 802.1Q Virtual LAN (0x8100)
    [Stream index: 0]
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 134
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 1000 0110 = ID: 134
    Type: IPv4 (0x0800)
    Trailer: 0000000000000000000000000000073636f00000000000000000000000000
Internet Protocol Version 4, Src: 62.0.134.2, Dst: 224.0.0.102
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        0000 00.. = Differentiated Services Codepoint: Default (0)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 80
    Identification: 0x0000 (0)
    000. .... = Flags: 0x0
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Protocol: UDP (17)
    Header Checksum: 0x1734 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 62.0.134.2
    Destination Address: 224.0.0.102
    [Stream index: 0]
User Datagram Protocol, Src Port: 1985, Dst Port: 1985
    Source Port: 1985
    Destination Port: 1985
    Length: 60
    Checksum: 0x5b42 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Stream Packet Number: 1]
    UDP payload (52 bytes)
Cisco Hot Standby Router Protocol
    Group State TLV: Type=1 Len=40
        Version: 2
        Op Code: Hello (0)
        State: Standby (5)
        IP Ver.: IPv4 (4)
        Group: 134
        Identifier: Cisco_21:e5:5b (40:14:82:21:e5:5b)
        Priority: 100
        Hellotime: Default (3000)
        Holdtime: Default (10000)
        Virtual IP Address: 62.0.134.3
    Text Authentication TLV: Type=3 Len=8
        Authentication Data: Default (cisco)

Packet Summary Decode:

Packet capture Summary : (Captured at RXPP) :

Ingress port details:
Interface                       :  Eth1/22
LTL                             :  0x58
System Port                     :  0x98
PIF                             :  24
Slice                           :  1
ifg                             :  1

Packet drop summary:
Packet dropped                  : NO

Packet punt summary:
Punt Details                    :Packet is not punted

Packet Details:
===============

decode_termination_input slice_idi 1

Packet Bytes(up to 128B) :
0x01005e00006640148221e55b81000086080045000050000000000ff1117343e008602e000006607c107c1003c5b42012802000504008640148221
e55b0000006400000bb8000027103e0086030000000000000000000000000308636973636f00000000000000000000000000000073636f000000000000000000000000

==== FI Array ====
------------------------------------------------------------------------------------------
FIELD_NAME                    :                            VALUE
------------------------------------------------------------------------------------------
array[9]                      :                              0x0
array[8]                      :                              0x0
array[7]                      :                              0x0
array[6]                      :                              0x0
array[5]                      :                              0x0
array[4]                      :                           0x2e00
```

```
array[3]                      :                            0x260f
  offset_in_bytes                                            0x26
  protocol_type                                 PROTOCOL_TYPE_UDP
array[2]                      :                            0x1214
  offset_in_bytes                                            0x12
  protocol_type                             PROTOCOL_TYPE_IPV4_L4
array[1]                      :                             0xe48
  offset_in_bytes                                             0xe
  protocol_type                                PROTOCOL_TYPE_VLAN
  flags                                                         2
array[0]                      :                              0x11
  offset_in_bytes                                             0x0
  protocol_type                       PROTOCOL_TYPE_ETHERNET_VLAN

==== RXPP Termination Input ====

------------------------------------------------------------------------------------------
FIELD_NAME                    :                             VALUE
------------------------------------------------------------------------------------------
padding_1                     :                               0x0
pch_label                     :                               0x0
unsch_rcy_code                :                               0x0
tx_to_rx_rcy_data             :                              0xf4
mtu_violation                 :                               0x0
initial_tc                    :                               0x0
offset_in_fragment            :                               0x0
slice_source_system_port      :                              0x98
processing_code               :                               0x0
destination                   :                               0x0
use_cache                     :                               0x0
single_fragment_packet        :                               0x1
flow_signature_on_npuh        :                               0x0
phb                           :                               0x0
reassembly_context            :                             0x7ff
learn_enable                  :                               0x0
receive_time_from_nppd        :                               0x0
rxnpu_recycle_count           :                               0x0
rxnpu_recycle_data            :                               0x0

==== RXPP Termination Macro Stack ====

NPE-macros-stack[0]: network_rx_mac_af_and_termination_macro
NPE-macros-stack[1]: network_rx_mac_relay_ipv4_mc_termination_macro

==== IRXPP Termination Lookup Keys/Results ====


NPE-lookup Keys/Results[0]:
no lookup hit, bucket #1 context network engine termination
no lookup hit, bucket #2 context network engine termination
+--------+--------------------------------------------+-----------------+---------+---------
--------------------------------------+--------------+
| Key Bucket |                                Key Type                               |
     Key Value          | Result Bucket |
      Result Type                     | Result Value |
+--------+--------------------------------------------+-----------------+---------+---------
--------------------------------------+--------------+
|     a      |          npl_service_mapping_ac_port_tag_compound_table_key_t         |
0x5e000066000000138184327c |       d       |
        NoneType                      |      0       |
|     d      | npl_ingress_qos_tag_encoding_pack_table_key_option_tag_type_v4_dscp_t |
        0x0             |       d       | npl_ingress_q
os_tag_encoding_pack_table_payloads_t |     0x0      |
+--------+--------------------------------------------+-----------------+---------+---------
```

```
--------------------------------------+--------------+

NPE-lookup Keys/Results[1]:
no lookup hit, bucket #0 context network engine termination
no lookup hit, bucket #2 context network engine termination
+--------+------------------------------------+-------+---------+--------------------------
-----------+------------------------------+
| Key Bucket |                         Key Type                         | Key Value |
Result Bucket |                    Result Type
           |         Result Value         |
+--------+------------------------------------+-------+---------+--------------------------
-----------+------------------------------+
|     b      | npl_mac_mc_em_termination_attributes_compound_table_key_t |   0x21b   |
   b       |             npl_base_l3_lp_attr_union_t
           | 0xe00010c32af008674000000000 |
|     d      |      npl_mc_macro_compressed_fileds_pack_table_key_t      |   0x388   |
   d       | npl_mc_macro_compressed_fileds_pack_table_
payloads_t |             0x0              |
+--------+------------------------------------+-------+---------+--------------------------
-----------+------------------------------+

==== RXPP Termination Output ====

------------------------------------------------------------------------------------------
FIELD_NAME                    :                              VALUE
------------------------------------------------------------------------------------------
learn_command                 :                                0x0
lb_command                    :                                0x0
offset_in_fragment            :                                0x0
slice_source_system_port      :                               0x98
processing_code               :                                0x0
destination                   :                            0xe0086
use_cache                     :                                0x0
single_fragment_packet        :                                0x1
flow_signature_on_npuh        :                                0x0
phb                           :                                0x0
reassembly_context            :                              0x7ff
learn_enable                  :                                0x0
receive_time_from_nppd        :                                0x0
rxnpu_recycle_count           :                                0x0
rxnpu_recycle_data            :                                0x0

==== RXPP Forwarding Macro Stack ====

NPE-macros-stack[0]: network_rx_ipv4_rtf_macro
NPE-macros-stack[1]: network_rx_mac_forwarding_macro
NPE-macros-stack[2]: resolution_macro

==== RXPP Forwarding Lookup Keys/Results ====


NPE-lookup Keys/Results[0]:
no lookup hit, bucket #1 context network engine forwarding
no lookup hit, bucket #3 context network engine forwarding
Error result bucket # 0 , from table ingress_rtf_ipv4_db1_240_f0_compound_table , overlapping
 previous value
Error result bucket # 1 , from table ingress_rtf_ipv4_db1_240_f0_compound_table , overlapping
 previous value
Error result bucket # 2 , from table ingress_rtf_ipv4_db1_240_f0_compound_table , overlapping
 previous value
Error result bucket # 3 , from table ingress_rtf_ipv4_db1_240_f0_compound_table , overlapping
 previous value
+--------+---------------------------------+--------------------------+---------+----------
-+--------------+
| Key Bucket |                       Key Type                       |                Key
Value                | Result Bucket |    Result Type
 | Result Value |
+--------+----------------------------------+--------------------------+----------+-----------
-+--------------+
|     a      | npl_ingress_rtf_ipv6_db4_480_f0_compound_table_key_t |
0x4ff45003e008602e000006607c19c0101f056 |       b       | npl_rtf_payload_t
 |     0x0      |
|     a      | npl_ingress_rtf_ipv6_db4_480_f0_compound_table_key_t |
0x4ff45003e008602e000006607c19c0101f056 |       c       | npl_rtf_payload_t
 |     0x0      |
|     a      | npl_ingress_rtf_ipv6_db4_480_f0_compound_table_key_t |
0x4ff45003e008602e000006607c19c0101f056 |       d       | npl_rtf_payload_t
 |     0x0      |
|     a      | npl_ingress_rtf_ipv6_db4_480_f0_compound_table_key_t |
0x4ff45003e008602e000006607c19c0101f056 |       a       | npl_rtf_payload_t
 |     0x0      |
|     c     | npl_ingress_rtf_ipv4_db1_240_f0_compound_table_key_t |                0x3821a
                 |       b       | npl_rtf_payload_t
 |     0x0      |
|     c     | npl_ingress_rtf_ipv4_db1_240_f0_compound_table_key_t |                0x3821a
                 |       c       | npl_rtf_payload_t
 |     0x0      |
|     c     | npl_ingress_rtf_ipv4_db1_240_f0_compound_table_key_t |                0x3821a
                 |       d       | npl_rtf_payload_t
 |     0x0      |
|     c     | npl_ingress_rtf_ipv4_db1_240_f0_compound_table_key_t |                0x3821a
                 |       a       | npl_rtf_payload_t
 |     0x0      |
+--------+----------------------------------+--------------------------+----------+-----------
-+--------------+

NPE-lookup Keys/Results[1]:
no lookup hit, bucket #0 context network engine forwarding
no lookup hit, bucket #1 context network engine forwarding
no lookup hit, bucket #2 context network engine forwarding
+--------+--------------------------+-------------+---------+-----------------------------+---
---------+
| Key Bucket |                Key Type                |     Key Value      | Result Bucket
 |                 Result Type                  | Resu
lt Value |
+--------+--------------------------+-------------+---------+-----------------------------+---
---------+
|     d      | npl_mac_forwarding_table_compound_key_t | 0x8601005e00006612 |       a
  | npl_mac_forwarding_table_compound_payloads_t |
0x0      |
+--------+--------------------------+-------------+---------+-----------------------------+---
---------+

NPE-lookup Keys/Results[2]:
no lookup hit, bucket #3 context network engine forwarding
+--------+-------------------------------------------+--------------------------+----------+--
----------------------------------------------------+--------------+
| Key Bucket |                               Key Type                               |
          Key Value                 | Result Bucket |
                 Result Type                      | Result Value |
+--------+-------------------------------------------+--------------------------+----------+--
----------------------------------------------------+--------------+
|     a      |              npl_v4_l4_resolution_table_compound_key_t              |
       0x3300103802183              |       a       |
     npl_resolution_table_compound_payloads_t       | 0xa9100e0086 |
|     b      |              npl_v4_l4_resolution_table_compound_key_t              |
0x113e008602e00007c107c1800000000000000 |       a       |
     npl_resolution_table_compound_payloads_t       | 0xa9100e0086 |
|     c      | npl_select_fwd_q_m_counter_base_pack_table_key_option_false_value_t |
          0x4000000                  |       c       | np
l_select_fwd_q_m_counter_base_pack_table_payloads_t |     0x0       |
+-------+-------------------------------------------+--------------------------+---------+--
----------------------------------------------------+--------------+

==== RXPP Forwarding Output ====

------------------------------------------------------------------------------------------
FIELD_NAME                    :                             VALUE
------------------------------------------------------------------------------------------
offset_in_fragment            :                               0x0
slice_source_system_port      :                              0x98
processing_code               :                               0x0
destination                   :                               0x0
use_cache                     :                               0x0
single_fragment_packet        :                               0x1
flow_signature_on_npuh        :                               0x0
phb                           :                               0x0
reassembly_context            :                             0x7ff
learn_enable                  :                               0x0
receive_time_from_nppd        :                               0x0
rxnpu_recycle_count           :                               0x0
rxnpu_recycle_data            :                               0x0
padding_2                     :                               0x0
use_ecn                       :                               0x0
fllb_control_code             :                               0x0
padding_1                     :                               0x0
ethernet_rate_limiter_type    :                               0x7
fwd_offset_cmd                :                               0x0

==== RXPP TM PD IFG0 ====

--------------------------------------------------------------------------------
FIELD_NAME                              :         VALUE
--------------------------------------------------------------------------------
color                                   :           0x0
counter_meter_command                   :      0xfdfcf5
is_dummy_pd                             :           0x1
reorder_data                            :        0xb72a
drop                                    :           0x1
forwarding_destination                  :           0x0
mirror_bitmap                           :        0x37ff
source_slice_system_port                :           0x0
traffic_class                           :           0x0
slice_mode_data                         :    0x1fb7ff4fcc9fd1ec2f4000000000000
processing_code                         :           0x0
lb_key_msbs_bits                        :           0x0
packet_size_bits                        :           0x0

==== RXPP TM PD IFG1 ====

--------------------------------------------------------------------------------
FIELD_NAME                              :         VALUE
--------------------------------------------------------------------------------
color                                   :           0x0
counter_meter_command                   :         0x43d
is_dummy_pd                             :           0x0
reorder_data                            :        0xba70
drop                                    :           0x0
forwarding_destination                  :      0x100086
mirror_bitmap                           :          0x10
source_slice_system_port                :          0x98
traffic_class                           :           0x0
```

```
slice_mode_data                         :    0x5ffffdfffffdfffffc0217227fffd0
processing_code                         :           0x0
lb_key_msbs_bits                        :           0x0
packet_size_bits                        :           0x0

==== NPU HEADER IFG0 ====

45ca99bdfecfefd5332fbfd000000000000000000000000000000000000000000000000000000000
------------------------------------------------------------------------------------------
FIELD_NAME                    :                             VALUE
------------------------------------------------------------------------------------------
base_type                     :                               0x4
version                       :                               0x0
ive_valid                     :                               0x1
packet_edit_valid             :                               0x0
issu_codespace                :                               0x1
receive_time                  :                        0xca99bdfe
meter_color                   :                               0x3
l2_flood_mc_pruning_or_etm    :                               0x0
ingress_qos_remark            :                           0xfefd5
fwd_header_type               :                               0x3
rx_nw_app_or_lb_key           :                             0x32f
fwd_offset                    :                              0xbf
slp_qos_id                    :                               0xd
encap_type                    :                               0x0
slp_dm_ptp                    :                               0x0
is_inject_packet_capture_en   :                               0x0
is_inject_up                  :                               0x0
ip_first_fragment             :                               0x0
ttl                           :                               0x0
collapsed_mc                  :                               0x0
da_bcast_or_mc_rpf            :                               0x0
slp_profile                   :                               0x0
l2_slp                        :                               0x0
l3_slp                        :                               0x0
is_l2                         :                               0x0
is_rpf_id                     :                               0x0
value                         :                               0x0
sgt                           :                               0x0

==== NPU HEADER IFG1 ====

04000000000200000044330f000000000000008600000008000010009e00100860000000000000058
------------------------------------------------------------------------------------------
FIELD_NAME                    :                             VALUE
------------------------------------------------------------------------------------------
base_type                     :                               0x0
version                       :                               0x0
ive_valid                     :                               0x1
packet_edit_valid             :                               0x0
issu_codespace                :                               0x0
receive_time                  :                               0x0
meter_color                   :                               0x0
l2_flood_mc_pruning_or_etm    :                               0x0
ingress_qos_remark            :                           0x20000
fwd_header_type               :                               0x0
rx_nw_app_or_lb_key           :                             0x443
fwd_offset                    :                              0x30
slp_qos_id                    :                               0xf
encap_type                    :                               0x0
L2_encap                      :                               0x0
padding                       :                               0x0
l2_dlp                        :                               0x0
pif                           :                               0x0
```

```
ifg                                :                           0x0
slp_dm_ptp                         :                           0x0
is_inject_packet_capture_en        :                           0x0
is_inject_up                       :                           0x0
ip_first_fragment                  :                           0x1
ttl                                :                           0x0
collapsed_mc                       :                           0x0
da_bcast_or_mc_rpf                 :                           0x0
slp_profile                        :                           0x0
l2_slp                             :                       0x9e001
l3_slp                             :                        0x9e00
is_l2                              :                           0x1
is_rpf_id                          :                           0x1
value                              :                       0x9e001
sgt                                :                           0x0

switch(config-pt-rxpp-pkt-fmt)#
```

Additional References

Related Documents	Title/Link
Cisco Nexus 9364E Switches Hardware Installation Guide	Cisco Nexus 9364E-SG2-Q Switch Hardware Installation Guide Cisco Nexus 9364E-SG2-O Switch Hardware Installation Guide
ELAM Overview	https://www.cisco.com/c/en/us/support/docs/switches/nexus-7000-series-switches/116648-technote-product-00.html
Nexus 9000 Cloud Scale ASIC (Tahoe) NX-OS ELAM	Nexus 9000 Cloud Scale ASIC (Tahoe) NX-OS ELAM - Cisco

CHAPTER 12

Troubleshooting PowerOn Auto Provisioning

Switch Does Not Come Up in Time for POAP to Complete

If the switch does not come up in a reasonable duration for POAP to complete, connect to the switch through the serial line and check to see if it is stuck at the following prompt:

```
Waiting for system online status before starting POAP ...
Waiting for system online status before starting POAP ...
Waiting for system online status before starting POAP ...

System is not fully online. Skip POAP? (yes/no)[n]:
```

You can continue with POAP by entering **no** at the prompt. If POAP does not start properly on the second attempt, proceed with the normal setup by entering **yes** at the prompt when it returns.

POAP Fails

Take these actions if any of the following PowerOn Auto Provisioning (POAP) errors appear:

Problem	Log Example	Solution
POAP does not get aborted or POAP abort is stuck at the “Disabling POAP” log	`Abort Auto Provisioning and continue with normal setup ?(yes/no)[n]: yes`	1. Use **Ctrl+c** or **Ctrl+z** to abort the POAP process and enter the switch. 2. If the above solution fails, power cycle the switch. 3. Abort POAP at an earlier prompt **Note** After aborting POAP and performing the necessary configurations or maintenance, you can save the configuration and reboot the switch to ensure it starts normally without entering POAP.

POAP DHCP offer is not accepted	`2022 Nov 17 11:55:59 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: [FOX2249PGK1-D4:C9:3C:85:7D:BF] - Missing Nexthop information, Option(242)`
	`2022 Nov 17 11:55:59 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: [FOX2249PGK1-D4:C9:3C:85:7D:BF] - Missing RT Prefix information, Option(243)`
	`2022 Nov 17 11:55:59 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: [FOX2249PGK1-D4:C9:3C:85:7D:BF] - Missing bootfile url, Option(59)`
		Add the prompted missing DHCP option as printed on the console to the DHCP server configuration.

Problem	Log Example	Solution

POAP script does not get copied	The error message is printed after "Copy Failed"
	`2022 Mar 10 22:46:52 switch %$ VDC-1 %$ %USER-1-SYSTEM_MSG: S/N[FDO25020XF4]-MAC[A0:3D:6E:EE:D8:40] - Command is : terminal dont-ask ; terminal password <removed> ; copy http://ecodemo.firmware/images/isan_nxos_9.3.8/nxos.9.3.8.bin bootflash:/nxos.9.3.8.bin.tmp vrf management - /script.sh`
	`2022 Mar 10 22:47:22 switch %$ VDC-1 %$ %USER-1-SYSTEM_MSG: S/N[FDO25020XF4]-MAC[A0:3D:6E:EE:D8:40] - Copy failed: "\nERROR: ld.so: object '/isan/lib/libutils.so' from LD_PRELOAD cannot be preloaded (wrong ELF class: ELFCLASS32): ignored.\nERROR: ld.so: object '/isan/lib/libsandbox.so' from LD_PRELOAD cannot be preloaded (wrong ELF class: ELFCLA`
		Make sure that the file name mentioned in the bootfile URL is correct, and the file is stored in the location as mentioned in the **copy** command output.

POAP script errors out with no error message	`178b171b535356627f7517e7a4c89d25 2022 Jun 9 00:17:55 switch %$ VDC-1 %$ %POAP-2-POAP_SCRIPT_STARTED_MD5_VALIDATED: [FOC232800YF-08:4F:A9:E4:95:37] - POAP script execution started(MD5 validated) 2022 Jun 9 00:17:56 switch %$ VDC-1 %$ %POAP-2-POAP_FAILURE: [FOC232800YF-08:4F:A9:E4:95:37] - POAP Script execution failed`	Run the python script on a Linux machine or Cisco Nexus switch using the **python3** command to capture the syntax errors. Once you've found the syntax error, resolve the error with the information provided.

Problem	Log Example	Solution
POAP script fails with error	2023 Apr 26 16:59:00 switch %$ VDC-1 %$ %USER-1-SYSTEM_MSG: - Executing configure terminal ; show crypto ca trustpoints - /script.sh^M^M 2023 Apr 26 16:59:01 switch %$ VDC-1 %$ %USER-1-SYSTEM_MSG: - Trustpoint already present. Please check. Exiting USB script. - /script.sh^M^M 2023 Apr 26 16:59:02 switch %$ VDC-1 %$ %POAP-2-POAP_FAILURE: [FDO25110HUV-F8:7A:41:55:30:9F] - POAP Script execution failed^M^M 2023 Apr 26 16:59:07 switch %$ VDC-1 %$ %POAP-2-POAP_FAILURE: [FDO25110HUV-F8:7A:41:55:30:9F] - POAP Script execution failed	Examine the specific error message that precedes this line “Script execution failed”. The error message will typically provide details about what went wrong and help you identify the part of the script that needs to be addressed. Resolve the error with the information provided in the logs.
Configuration is missing after POAP replay	root@switch(config)# route-map test % Incomplete command at ‘^’ marker ret=-19	Use the **show startup-config poap-log** command and check for the missing configuration. Configure the missing configuration until the issue is resolved.

- Stop the POAP process to continue with the normal switch bring-up steps. It might take a few minutes for a full stop of POAP, so be patient.

```
2013 Oct 29 22:24:59 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Assigned IP address:
172.23.40.221
2013 Oct 29 22:24:59 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Netmask: 255.255.255.0
2013 Oct 29 22:24:59 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: DNS Server: 172.21.157.5
2013 Oct 29 22:24:59 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Default Gateway: 172.23.40.1
2013 Oct 29 22:24:59 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Script Server: 172.23.40.6
2013 Oct 29 22:24:59 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Script Name: /pxelinux.0
2013 Oct 29 22:25:09 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: The POAP Script download has
 started
2013 Oct 29 22:25:09 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: The POAP Script is being
downloaded from [copy tftp://172.23.40.6//pxelinux.0 bootflash:scripts/script.sh vrf
management ]
2013 Oct 29 22:25:10 switch %$ VDC-1 %$ %POAP-2-POAP_FAILURE: POAP boot file download
```

```
failed.
2013 Oct 29 22:25:10 switch %$ VDC-1 %$ %POAP-2-POAP_FAILURE: POAP DHCP discover phase
 failed
2013 Oct 29 22:25:12 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Abort Power On Auto
Provisioning and continue with normal setup ?(yes/no)[n]:
2013 Oct 29 22:25:46 switch %$ VDC-1 %$ %POAP-2-POAP_DHCP_DISCOVER_START: POAP DHCP
Discover phase started
2013 Oct 29 22:25:46 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Abort Power On Auto
Provisioning and continue with normal setup ?(yes/no)[n]:

Abort Auto Provisioning and continue with normal setup ?(yes/no)[n]: yes
```

- Check the failure reason in the log files. Two POAP log files are saved on the bootflash. Logs from the POAP process are stored in the file that ends with poap_*pid*_init.log as shown below. The failure reason should appear toward the end of this file.

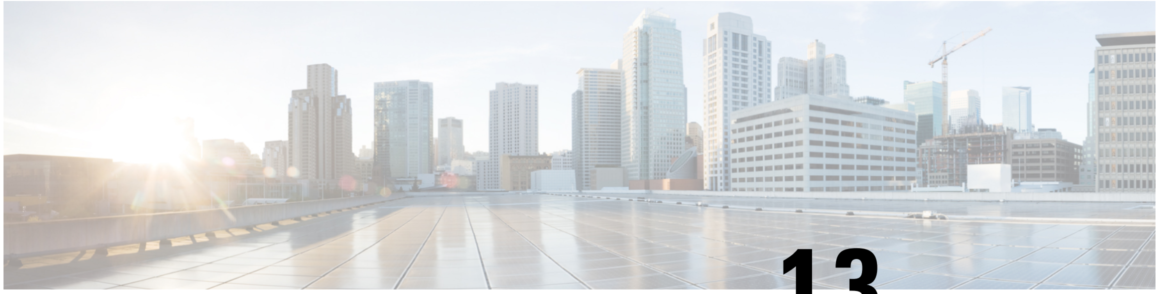
```
bash-4.2# tail 20131029_222312_poap_5367_init.log -n 3
Tue Oct 29 22:27:41 2013:poap_net_rx_pkt: Droppping the pakcet due to Ethernet hdrparsing
 error on if_index - 5000000
Tue Oct 29 22:27:41 2013:DEST IP is not Broadcast
Tue Oct 29 22:27:41 2013:poap_net_rx_pkt: Droppping the pakcet due to Ethernet hdrparsing
 error on if_index - 5000000
```

- Check to see if the POAP script file that is downloaded from your DHCP or TFTP server fails in the process of running. Depending on the stage of the failure, the device might proceed with the normal setup or reboot.

```
2013 Oct 29 22:42:34 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Assigned IP address:
172.23.40.181
2013 Oct 29 22:42:34 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Netmask: 255.255.255.0
2013 Oct 29 22:42:34 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: DNS Server: 172.21.157.5
2013 Oct 29 22:42:34 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Default Gateway: 172.23.40.1
2013 Oct 29 22:42:34 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Script Server: 172.23.40.6
2013 Oct 29 22:42:34 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Script Name: poap.py
2013 Oct 29 22:42:45 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: The POAP Script download has
 started
2013 Oct 29 22:42:45 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: The POAP Script is being
downloaded from [copy tftp://172.23.40.6/poap.py bootflash:scripts/script.sh vrf
management ]
2013 Oct 29 22:42:46 switch %$ VDC-1 %$ %POAP-2-POAP_SCRIPT_DOWNLOADED: Successfully
downloaded POAP script file
2013 Oct 29 22:42:46 switch %$ VDC-1 %$ %POAP-2-POAP_INFO: Script file size 21965, MD5
 checksum 1bd4b86892439c5785a20a3e3ac2b0de
2013 Oct 29 22:42:46 switch %$ VDC-1 %$ %POAP-2-POAP_SCRIPT_STARTED_MD5_NOT_VALIDATED:
 POAP script execution started(MD5 not validated)
2013 Oct 29 22:47:57 switch %$ VDC-1 %$ %POAP-2-POAP_FAILURE: POAP script execution
aborted
```

- The POAP script file logs are written to a file under the bootflash scheme. The filename starts with poap.log. If there are multiple file logs, look at the one with the most recent time stamp for any errors.

```
bash-4.2# tail poap.log.22_42_46
CLI : show file volatile:poap.cfg.md5.poap_md5 | grep -v '^#' | head lines 1 | sed 's/
 .*$//'
INFO: md5sum 46684d8f8b7c5ffac3b37ac8560928e5 (.md5 file)
CLI : show file volatile:poap.cfg md5sum
INFO: md5sum 46684d8f8b7c5ffac3b37ac8560928e5 (recalculated)
CLI : show system internal platform internal info | grep box_online | sed 's/[^0-9]*//g'
INFO: Setting the boot variables
CLI : config terminal ; boot nxos bootflash:poap/system.img
CLI : copy running-config startup-config
CLI : copy volatile:poap.cfg scheduled-config
INFO: Configuration successful
```



CHAPTER 13

Troubleshooting the Python API

Receiving Python API Errors

Take these actions if any of the following Python API errors appear:

Symptom	Solution	Example

The Python cli API throws a NameError.	Import the cli module into the global namespace.	`>>> cli('show clock')`
		`Traceback (most recent call last):`
		`  File "<stdin>", line 1, in <module>`
		`NameError: name 'cli' is not defined`
		
		`>>> from cli import *`
		`>>> cli('show clock')`
		`'20:23:33.967 UTC Fri Nov 01 2013\n'`

The Python clid API throws a structured_output_not_supported_error.	Use the cli or clip API. The clid API works only with commands that support structured data output.	`>>> clid('show clock')`
		`Traceback (most recent call last):`
		`  File "<stdin>", line 1, in <module>`
		`  File "/isan/python/scripts/cli.py", line 45,`
		` in clid`
		`    raise`
		`structured_output_not_supported_error(cmd)`
		`errors.structured_output_not_supported_error:`
		` 'show clock'`

Symptom	Solution	Example

The cli API and cisco objects throw a Permission denied error.	Make sure your login ID has sufficient permissions to access the command or resource. If necessary, ask your network administrator for additional permissions.	>>> from cli import *
		>>> cli('clear counters')
		Traceback (most recent call last):
		  File "<stdin>", line 1, in <module>
		  File "/isan/python/scripts/cli.py", line 20,
		 in cli
		    raise cmd_exec_error(msg)
		errors.cmd_exec_error: '% Permission denied
		for the role\n\nCmd exec error.\n'
		>>> from cisco.interface import *
		>>> i=Interface('Ethernet3/2')
		Traceback (most recent call last):
		  File "<stdin>", line 1, in <module>
		  File
		"/isan/python/scripts/cisco/interface.py", line
		 75, in __new__
		    cls._Interfaces[name].config(True)
		  File
		"/isan/python/scripts/cisco/interface.py", line
		 91, in config
		    s, o = nxcli('show runn interface %s' %
		self.name)
		  File "/isan/python/scripts/cisco/nxcli.py",
		 line 46, in nxcli
		    raise SyntaxError, 'Error status %d\n%s'
		% (status, output)
		SyntaxError: Error status 30
		% Permission denied for the role
		
		Cmd exec error.
		
		>>> import os
		>>> os.system('whoami')
		test

Symptom	Solution	Example

The urllib2 or socket connection is not processed.	Make sure you are using the correct virtual routing context. If not, switch to the correct one.	>>> import urllib2
		>>>
		u=urllib2('http://172.23.40.211:8000/welcome.html')
		Traceback (most recent call last):
		  File "<stdin>", line 1, in <module>
		TypeError: 'module' object is not callable
		>>>
		u=urllib2.urlopen('http://172.23.40.211:8000/welcome.html')
		Traceback (most recent call last):
		  File "<stdin>", line 1, in <module>
		  File "/isan/python/python2.7/urllib2.py",
		line 127, in urlopen
		    return _opener.open(url, data, timeout)
		  File "/isan/python/python2.7/urllib2.py",
		line 404, in open
		    response = self._open(req, data)
		  File "/isan/python/python2.7/urllib2.py",
		line 422, in _open
		    '_open', req)
		  File "/isan/python/python2.7/urllib2.py",
		line 382, in _call_chain
		    result = func(*args)
		  File "/isan/python/python2.7/urllib2.py",
		line 1214, in http_open
		    return self.do_open(httplib.HTTPConnection,
		 req)
		  File "/isan/python/python2.7/urllib2.py",
		line 1184, in do_open
		    raise URLError(err)
		urllib2.URLError: <urlopen error [Errno 113]
		No route to host>
		>>> from cisco.vrf import *
		>>> VRF.get_vrf_name_by_id(get_global_vrf())
		'default'

CHAPTER 14

Troubleshooting NX-API

NX-API Guidelines

NX-API performs authentication through a programmable authentication module (PAM) on the switch. Use cookies to reduce the number of PAM authentications and thus reduce the load on PAM.

NX-API Is Not Responding

Take these actions if NX-API is not responding:

- Make sure that NX-API is enabled by using the **show feature | grep nxapi** command.
- Make sure that HTTP or HTTPs is enabled by using the **show nxapi** command.
- Make sure that NX-API is listening on the expected port by using the **show nxapi** command.
- Check for a long running command. Currently NX-API runs on a single worker process and is single threaded. If one command takes a long time to complete, it will block other commands. NX-API caches the request. When the current request completes, the others will be served.
- Enable Bash. For instructions, see the *Cisco Nexus 9000 Series NX-OS Programmability Guide*.
- Check the /var/sysmgr_nxapi/logs/error.log to see if there are any errors.
- If NX-API is still not responding, enter the **no feature nxapi** and **feature nxapi** commands to restart NX-API. NX-API is stateless, and it is safe to restart.

Configuration Fails

Take these actions if the user cannot execute configuration commands:

- Make sure that the user has the correct privileges to execute the commands.

Permission Is Denied for Bash

Take these actions if users receive a "Permission Denied" message for Bash:

- Make sure that Bash is enabled by using the **show feature | grep bash** command.
- Make sure that the current user has the correct privileges to access Bash.
- For more information on Bash, see the *Cisco Nexus 9000 Series NX-OS Programmability Guide*.

Output Cannot Be Retrieved from the Browser Sandbox

Take these actions if you cannot retrieve the output from the browser sandbox:

- When the output is large or the command execution takes a long time, the browser might not be able to handle the load and might time out. Try using the Python client to access the NX-API. For instructions, see the *Cisco Nexus 9000 Series NX-OS Programmability Guide*.

Note The recommended browser is Mozilla Firefox.

CLI Command Errors Are Appearing

Take these actions if CLI command errors appear when the user runs multiple commands:

- Check to see how multiple commands are separated. Show and configure commands must be separated by a [space]. Bash commands must be separated by a semicolon (;).

Error Messages Are Appearing

Take these actions if error messages are appearing in the output:

- Follow the instructions in the error message.
- If the Bash commands do not go through, make sure that Bash is enabled by using the **show feature | grep bash** command. For more information on Bash, see the *Cisco Nexus 9000 Series NX-OS Programmability Guide*.
- Make sure that the user has the correct privileges to execute the command.

- Follow the instructions in NX-API Is Not Responding, on page 129.

Temporary Files Are Disappearing

For every request, a temporary file is created in /volatile to store the command output that is sent back to the client. If the chunk parameter on the request is 0, the file is deleted right before the command output is sent back to the client. If the request does have chunk = 1, the file is retained so that the chunks can be extracted from it and sent to the client. That file will be cleaned up on a periodic basis. Currently that cleanup is set to occur once every 100 requests. Files are cleaned up if they are not accessed within 60 seconds of being created or are not modified or their status is not updated within 600 seconds.

Chunks of the Command Output Are Not Being Delivered

For requests where chunk = 1, if the sid is set to the same value, you will get the same chunk of the command output. This functionality allows for situations where a client requests a specific chunk and does not receive it in a timely manner because it is dropped or blocked somewhere in the network. The clients can request the same chunk again, and they will receive the correct data as long as the temporary file has not been cleaned up (as described in Temporary Files Are Disappearing, on page 131).

CHAPTER 15

Troubleshooting Service Failures

Identifying Memory Allocations for Processes

You can identify the allocation, limit, memory allocation, and usage for each process in the memory. The following is a sample output from the **show processes memory** command. This output has been abbreviated to make the example more concise.

```
switch# show processes memory
 PID  MemAlloc MemLimit    MemUsed     StackBase/Ptr      Process
 ---- -------- --------    -------     -----------------  -----------
   1    159744  0           2027520     ff808d30/ffffffff  init
   2         0  0                 0                   0/0  kthreadd
   3         0  0                 0                   0/0  migration/0
   4         0  0                 0                   0/0  ksoftirqd/0
   5         0  0                 0                   0/0  watchdog/0
   6         0  0                 0                   0/0  migration/1
   7         0  0                 0                   0/0  ksoftirqd/1
   8         0  0                 0                   0/0  watchdog/1
   9         0  0                 0                   0/0  migration/2
  10         0  0                 0                   0/0  ksoftirqd/2
  11         0  0                 0                   0/0  watchdog/2
  12         0  0                 0                   0/0  migration/3
  13         0  0                 0                   0/0  ksoftirqd/3
  14         0  0                 0                   0/0  watchdog/3
  15         0  0                 0                   0/0  migration/4
  16         0  0                 0                   0/0  ksoftirqd/4
  17         0  0                 0                   0/0  watchdog/4
  18         0  0                 0                   0/0  migration/5
  19         0  0                 0                   0/0  ksoftirqd/5
  20         0  0                 0                   0/0  watchdog/5
  21         0  0                 0                   0/0  migration/6
  22         0  0                 0                   0/0  ksoftirqd/6
  23         0  0                 0                   0/0  watchdog/6
  24         0  0                 0                   0/0  migration/7
  25         0  0                 0                   0/0  ksoftirqd/7
  26         0  0                 0                   0/0  watchdog/7
  27         0  0                  0                   0/0  events/0
  28         0  0                  0                   0/0  events/1
  29         0  0                  0                   0/0  events/2
  30         0  0                  0                   0/0  events/3
  31         0  0                  0                   0/0  events/4
  32         0  0                  0                   0/0  events/5
  33         0  0                  0                   0/0  events/6
  34         0  0                  0                   0/0  events/7
  35         0  0                  0                   0/0  khelper
  36         0  0                  0                   0/0  netns
  37         0  0                  0                   0/0  kblockd/0
```

The **show processes memory** command includes the following keywords:

Keyword	Description
>	Redirects the output to a file.
>>	Adds the output to an existing file.
shared	Displays shared memory information.

Identifying CPU Utilization for Processes

You can identify the CPU utilization for running process in the memory. The following is a sample output from the **show processes cpu** command. This output has been abbreviated to make the example more concise.

```
switch# show processes cpu

CPU utilization for five seconds: 0%/0%; one minute: 1%; five minutes: 2%

   PID    Runtime(ms)Invoked   uSecs 5Sec    1Min   5Min    TTY  Process
   ---    ------------------   ----- -----   -----  -----   ---  ----------
   1        28660    405831      70   0.00%   0.00%  0.00%   -    init
   2           21      1185      18   0.00%   0.00%  0.00%   -    kthreadd
   3          468     36439      12   0.00%   0.00%  0.00%   -    migration/0
   4        79725   8804385       9   0.00%   0.00%  0.00%   -    ksoftirqd/0
   5            0         4      65   0.00%   0.00%  0.00%   -    watchdog/0
   6          472     35942      13   0.00%   0.00%  0.00%   -    migration/1
   7        33967    953376      35   0.00%   0.00%  0.00%   -    ksoftirqd/1
   8            0        11       3   0.00%   0.00%  0.00%   -    watchdog/1
   9          424     35558      11   0.00%   0.00%  0.00%   -    migration/2
  10        58084   7683251       7   0.00%   0.00%  0.00%   -    ksoftirqd/2
  11            0         3       1   0.00%   0.00%  0.00%   -    watchdog/2
  12          381     29760      12   0.00%   0.00%  0.00%   -    migration/3
  13        17258    265884      64   0.00%   0.00%  0.00%   -    ksoftirqd/3
  14            0         2       0   0.00%   0.00%  0.00%   -    watchdog/3
  15        46558   1300598      35   0.00%   0.00%  0.00%   -    migration/4
  16      1332913   4354439     306   0.00%   0.00%  0.00%   -    ksoftirqd/4
  17            0         6       2   0.00%   0.00%  0.00%   -    watchdog/4
  18        45808   1283581      35   0.00%   0.00%  0.00%   -    migration/5
  19       981030   1973423     497   0.00%   0.00%  0.00%   -    ksoftirqd/5
  20            0        16       3   0.00%   0.00%  0.00%   -    watchdog/5
  21        48019   1334683      35   0.00%   0.00%  0.00%   -    migration/6
  22      1084448   2520990     430   0.00%   0.00%  0.00%   -    ksoftirqd/6
  23            0        31       3   0.00%   0.00%  0.00%   -    watchdog/6
  24        46490   1306203      35   0.00%   0.00%  0.00%   -    migration/7
  25      1187547   2867126     414   0.00%   0.00%  0.00%   -    ksoftirqd/7
  26            0        16       3   0.00%   0.00%  0.00%   -    watchdog/7
  27        21249   2024626      10   0.00%   0.00%  0.00%   -    events/0
  28         8503   1990090       4   0.00%   0.00%  0.00%   -    events/1
  29        11675   1993684       5   0.00%   0.00%  0.00%   -    events/2
  30         9090   1973913       4   0.00%   0.00%  0.00%   -    events/3
  31        74118   2956999      25   0.00%   0.00%  0.00%   -    events/4
  32        76281   2837641      26   0.00%   0.00%  0.00%   -    events/5
  33       129651   3874436      33   0.00%   0.00%  0.00%   -    events/6
  34         8864   2077714       4   0.00%   0.00%  0.00%   -    events/7
  35            0         8      23   0.00%   0.00%  0.00%   -    khelper
  36          234        34    6884   0.00%   0.00%  0.00%   -    netns
```

The **show processes cpu** command includes the following keywords:

Keyword	Description
>	Redirects the output to a file.
>>	Adds the output to an existing file.
history	Displays information about the CPU utility.
sort	Sorts the list based on the memory usage.

Monitoring Process Core Files

You can monitor the process core files by using the **show cores** command.

```
switch# show cores
Module  Instance  Process-name     PID       Date(Year-Month-Day Time)
------  --------  ---------------  --------  -------------------------
28      1         bgp-64551        5179      2013-11-08 23:51:26
```

The output shows all cores that are presently available for upload from the active supervisor.

Processing the Crash Core Files

You can process the crash core files by using the **show processes log** command.

```
switch# show process log
  Process           PID     Normal-exit  Stack-trace  Core     Log-create-time
  ----------------  ------  -----------  -----------  -------  ---------------
  ntp               919               N            N        N  Jun 27 04:08
  snsm              972               N            Y        N  Jun 24 20:50
```

Clearing the Core

You can clear the core by using the **clear cores** command.

```
switch# clear cores
```

Enabling Auto-Copy for Core Files

You can enter the **system cores** command to enable the automatic copy of core files to a TFTP server, the flash drive, or a file.

```
switch(config)# system cores tftp://10.1.1.1/cores
```

CHAPTER 16

Before Contacting Technical Support

Steps to Perform Before Calling TAC

At some point, you might need to contact your technical support representative or Cisco TAC for some additional assistance. This section outlines the steps that you should perform before you contact your next level of support in order to reduce the amount of time spent resolving the issue.

To prepare for contacting your customer support representative, follow these steps:

1. Collect the system information and configuration. You should collect this information before and after the issue has been resolved. Use one of the following three methods to gather this information:

   - Configure your Telnet or Secure Shell (SSH) application to log the screen output to a text file. Use the **terminal length 0** command and then use the **show tech-support details** command.

Note If certain **show tech** commands generate a large amount of data and occupy more disk space, they can be stored in a compressed format. See the following example:

```
bash-4.2# time vsh -c " show tech-support platform-sdk" | gzip >
/bootflash/pltfm-tech.gz
```

Note SSH timeout period must be longer than the time of the tac-pac generation time. Otherwise, the VSH log might show %VSHD-2-VSHD_SYSLOG_EOL_ERR error. Ideally, set to 0 (infinity) before collecting tac-pac or showtech.

   - Beginning with Cisco NX-OS Release 9.3(1), you can use the **show tech-support details** [**space-optimized** | **time-optimized**] command. The multi-threaded virtual shell can run up to 16 threads in parallel and monitor them at the same time. The space-optimized parameter removes duplicate input commands and zips the output to optimize memory utilization.

Note This command is not supported on devices with less than 4GB of RAM.

- Use the **tac-pac** *filename* command to redirect the output of the **show tech-support details** command to a file, and then gzip the file.

```
switch# tac-pac bootflash://showtech.switch1
```

- If you do not specify a filename, Cisco NX-OS creates the file as volatile:show_tech_out.gz. Copy the file from the device using the procedure in Copying Files to or from Cisco NX-OS, on page 139.

2. If an error occurs in DCNM, take a screen shot of the error. In Windows, press **Alt+PrintScreen** to capture the active window, or press **PrintScreen** to capture the entire desktop. Paste the screenshot into a new Microsoft Paint (or similar program) session and save the file.

3. Capture the exact error codes that you see in the message logs from either DCNM or the CLI.
   - Choose **Event Browser** in DCNM to see the recent list of messages generated.
   - Copy the error from the message log, which you can display by using either the **show logging logfile** or the **show logging last** *number* command to view the last lines of the log.

4. Answer the following questions before you contact your technical support representative:
   - On which device or port is the problem occurring?
   - Which Cisco NX-OS software, driver versions, operating systems versions, and storage device firmware are in your network?
   - What is the network topology? (In DCNM, choose **Topology** > **Save layout**.)
   - Were any changes made to the environment (VLANs, upgrades, or adding modules) 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 device 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 during the problem time? 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
     - DCNM tools

5. Answer the following questions if your problem is related to a software upgrade attempt:
   - What was the original Cisco NX-OS version?
   - What is the new Cisco NX-OS version?
   - Collect the output from the following commands and forward them to your customer support representative:
     - **show install all status**
     - **show system internal log install**
     - **show system internal log install details**
     - **show log nvram**
6. Beginning with Cisco NX-OS Release 10.3.1(F), a new CLI **slot X show hardware internal statistics all** is added to collect hardware statistics for all the slots (TOR/EOR).
7. The following is the list of CLIs that are added in the **show-tech support module all** command:
   - Slot XX **show hardware internal buffer info pkt-stats input instance <ASIC>**
   - Slot XX **show hardware internal jer-usd stats interrupt asic <ASIC>**
   - Slot XX **show hardware internal jer-usd stats traffic-rate asic <ASIC>**
   - Slot XX **show hardware internal jer-usd stats port-queue front-port <front_port_number>**
   - Slot XX **show hardware internal buffer info pkt-stats input instance <ASIC>**
   - Slot XX **show hardware internal jer-usd stats vsq front-port <front_port_number>**
   - Slot XX **show hardware internal jer-usd stats vsq inband asic <ASIC>**

   The following is the information on the keywords used in the above commands:

Keywords	Description
XX	Module number
ASIC	ASIC number supported by platform
<front_port_number>	Port number range supported by the platform

Copying Files to or from Cisco NX-OS

You might need to move files to or from the device. These files may include the log, configuration, or firmware files.

Cisco NX-OS offers protocols to use for copying to or from the device. The device always acts as a client, so that an FTP, SCP, or TFTP session always originates from Cisco NX-OS 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** command supports the FTP, SCP, SFTP, and TFTP transfer protocols and many different sources for copying files.

```
switch# copy ?
 bootflash:      Select source filesystem
 core:           Select source filesystem
 debug:          Select source filesystem
 ftp:            Select source filesystem
 http:           Select source filesystem
 https:          Select source filesystem
 licenses        Backup license files
 log:            Select source filesystem
 logflash:       Select source filesystem
 nvram:          Select source filesystem
 running-config  Copy running configuration to destination
 scp:            Select source filesystem
 sftp:           Select source filesystem
 startup-config  Copy startup configuration to destination
 system:         Select source filesystem
 tftp:           Select source filesystem
 usb1:           Select source filesystem
 usb2:           Select source filesystem
 volatile:       Select source filesystem
```

You can use secure copy (SCP) as the transfer mechanism, as follows:

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

This example copies /etc/hosts from 172.22.36.10 to hosts.txt, for user user1:

```
switch# 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 backs up the startup configuration to an SFTP server:

```
switch# 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:
switch#
```

Note You should back up the startup configuration to a server on a daily basis and prior to any changes. You could write a short script to run on Cisco NX-OS to perform a save and then a backup of the configuration. The script needs to contain two commands: **copy running-configuration startup-configuration** and **copy startup-configuration tftp:***//server/name*. To execute the script, use the **run-script** *filename* command.

Custom Port in Copy Command

The following command enables you to specify port numbers for SCP or SFTP and other protocols such as HTTPS, TFTP, and FTP. This command can be used to copy files from/to an Nexus switch where the existing copy protocols are running on custom ports.

```
switch# copy <scheme>://[username @]hostname/filepath directory port <port-number>
```

Using Core Dumps

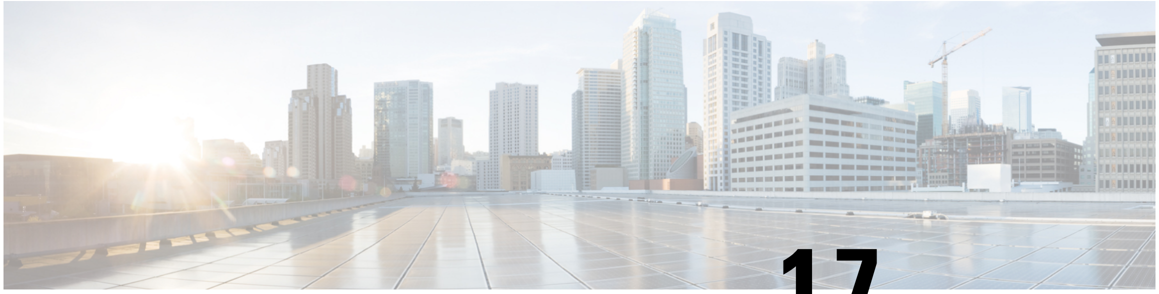
Core dumps contain detailed information about the system and software status prior to a crash. Use core dumps in situations where unknown problems exist. You can send core dumps to a TFTP server or to a Flash card in slot0: of the local system. You should set up your system to generate core dumps under the instruction of your technical support representative. Core dumps are decoded by technical support engineers.

Set up core dumps to go to a TFTP server so that you can e-mail these core dumps directly to your technical support representative.

Use the **system cores** command to set up core dumps on your system as follows:

```
switch# system cores tftp://10.91.51.200/jsmith_cores
switch# 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.



CHAPTER 17

Troubleshooting Tools and Methodology

Command-Line Interface Troubleshooting Commands

The command-line interface (CLI) allows you to configure and monitor Cisco NX-OS using a local console or remotely using a Telnet or Secure Shell (SSH) session. The CLI provides a command structure similar to Cisco IOS software, with context-sensitive help, **show** commands, multiuser support, and roles-based access control.

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

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

```
switch# copy running-config startup-config
[########################################] 100%
2013 May 16 09:59:29 zoom %$ VDC-1 %$ %BOOTVAR-2-AUTOCOPY_FAILED: Autocopy of file
/bootflash/n9000-dk9.6.1.2.I1.1.bin to standby

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

Consistency Checker Commands

Cisco NX-OS provides consistency checker commands to validate the software state with the hardware state. The result of the consistency checker is logged as either PASSED or FAILED.

```
2019 May 1 16:31:39 switch vshd: CC_LINK_STATE:
Consistency Check: PASSED
```

Consistency checker is a tool that performs the following functions:

- Checks for system consistency
- Helps perform root cause analysis and fault isolation
- Checks for consistency between software and hardware tables

Note When monitor session is in Down or Error state, Consistency checker is not validated.

Cisco NX-OS supports the following consistency checker commands.

Table 4: Consistency Checker Commands

Command	Description	Supported Platforms
show consistency-checker copp	Verifies CoPP programming.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards Cisco Nexus N9K-C9316D-GX, N9K-C93600CD-GX, N9K-C9364C-GX switches

Command	Description	Supported Platforms
show consistency-checker copp extended module *module_no* [**brief** | **detail**] **Note** Beginning with Cisco NX-OS Release 10.5(3)F, this command is deprecated.	Verifies the consistency of all control-plane ACLs. - **Brief** – Displays Output of failed entries in structured format. **Detail** - Displays Output of all ACE entries in structured format.	Cisco Nexus 9300-FX3/GX/GX2/H2R/H1, 9808, and 9804 Series switches.
show consistency-checker control-plane acl extended module *module_no* [**brief** | **detail**]	Verifies the consistency of all control-plane ACLs. This new command is enhanced with more qualifiers to perform the CoPP check more effectively.	Cisco Nexus 9300-FX/FX2/FX3/GX/GX2/H2R/H1, 9808, and 9804 Series switches, and 9500 Series switches with 9700- EX/FX/FX3/GX line cards
show consistency-checker dme interfaces	Verifies the DME interfaces.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards
show consistency-checker egress-xlate private-vlan	Verifies the private VLAN egress-xlate in the hardware.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards
show consistency-checker fex-interfaces {**fex** *fex-id* | **interface ethernet** *fex-id/fex-slot/fex-port*} [**brief** | **detail**]	Compares the software and hardware state of FEX interfaces.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards **Note** The *fex-slot* is always 1.
show consistency-checker fex-interfaces fabric <*fabric-po*>	Verifies FEX fabric PO membership of the physical member interfaces, and interface level hardware programming of the fabric port-channel members.	Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-GX series switches.
show consistency-checker fex-interfaces fabric <*fabric-po*> **membership vlan** <*vlan-id*>	Verifies FEX fabric PO members are part of the VLAN floodlist, for VLAN which are enabled on FEX interfaces.	Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-GX series switches.

Command	Description	Supported Platforms
show consistency-checker fex-interfaces fabric <*fabric-po*> **stp-state vlan** <*vlan-id*>	Verifies FEX fabric PO members are in forwarding / disabled state for VLANs that are enabled on FEX interfaces.	Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-GX series switches.
show consistency-checker fex-interfaces fabric <*fabric-po*> **egress-xlate private-vlan** <*vlan-id*>	Verifies PVLAN hardware programming corresponding to FEX fabric PO interface, in case there are PVLAN enabled FEX interfaces.	Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-GX series switches.
test consistency-checker forwarding {**ipv4** | **ipv6**} [**vrf** *vrf-name* | **all**] [**module** *module-number* | **all**]	Starts the Layer 3 route consistency checker test.	All Cisco Nexus 9000 Series switches
show consistency-checker forwarding {**ipv4** | **ipv6**} [**vrf** *vrf-name* | **all**] [**module** *module-number* | **all**]	Displays the Layer 3 route consistency checker test result.	All Cisco Nexus 9000 Series switches
show consistency-checker forwarding single-route {**ipv4** | **ipv6**} *ip-address* **vrf** *vrf-name*} [**brief** | **detail**]	Checks for Layer 3 route consistency for a specific route. Warns when a single-route fails due to ECMP group table exhaustion.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards **Note** Cisco Nexus 34180YC platform switches support only the **ipv4** command.
show consistency-checker gwmacdb	Checks for hardware and software consistency in the gateway MAC address database. **Note** This command might show incorrect results for 4-way HSRP.	All Cisco Nexus 9000 Series switches
show consistency-checker kim interface {**ethernet** *slot/port* | **port-channel** *number* | **vlan** *vlan-id*} [**brief** | **detail**]	Verifies the internal connectivity between the supervisor and the line card.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards
show consistency-checker l2 module *module-number*	Verifies that learned MAC addresses are consistent between the software and the hardware. It also shows extra entries that are present in the hardware but not in the software and missing entries in the hardware.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards

Command	Description	Supported Platforms
show consistency-checker l2 multicast group *ip-address* **source** *ip-address* **vlan** *vlan-id* [**brief** | **detail**]	Checks for inconsistencies with Layer 2 multicast groups.	Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-GX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards Cisco Nexus 9500 series switches with N9K-X9432C-S, N9K-X9536PQ line cards. Cisco Nexus 9500 series switches with N9K-X9432C-FM-S, N9K-C9508-FMX-S, N9K-C9508-FM-S fabric modules. Cisco Nexus N3K-C3232C, N3K-C3264Q, N3K-C31108TC-V, N3K-C3132Q-40GX, N3K-C3132Q-V, N3K-C31108PC-V, N3K-C3172PQ, N3K-C3172TQ, N3K-C3164Q, and N3K-C31128PQ-10GE switches. Cisco Nexus N9K-C9372TX, N9K-C9372TX-E, N9K-C93120TX, N9K-X9432C-S, N9K-C9332PQ, N9K-C9372PX and N9K-C9372PX-E switches.
show consistency-checker l2 switchport interface {**ethernet** *slot/port* | **port-channel** *number* }[**brief** | **detail** | **all**]	Checks for inconsistencies with switchport interfaces.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards

Command	Description	Supported Platforms
show consistency-checker l3-interface interface ethernet *slot/port* [**brief** | **detail**]	Checks for Layer 3 settings of an interface in the hardware and for the following configuration in the hardware: L3 VLAN, CML Flags, IPv4 Enable, VPN ID. This command works for physical interfaces and interfaces that are part of a port channel. It does not validate subinterfaces or FEX interfaces. Beginning Cisco NX-OS Release 9.3(5) this command checks for Layer 3 settings of an SI and SVI interfaces; and the support is extend to Cisco Nexus 9300-GX platform switches.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards Cisco Nexus N9K-C9316D-GX, N9K-C93600CD-GX, N9K-C9364C-GX devices. Beginning with Cisco NX-OS Release 10.3(1)F, L3 Consistency Checker is supported on the Cisco Nexus 9808 platform switches. Beginning with Cisco NX-OS Release 10.4(1)F, L3 Consistency Checker is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches. Beginning with Cisco NX-OS Release 10.4(1)F, L3 Consistency Checker is supported on the Cisco Nexus 9804 platform switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards. Beginning with Cisco NX-OS Release 10.4(2)F, L3 Consistency Checker is supported on the Cisco Nexus C9232E-B1 switch.
show consistency-checker l3-interface module *module-number* [**brief** | **detail**]	Checks for Layer 3 settings of all interfaces in the module and for the following configuration in the hardware: L3 VLAN, CML Flags, IPv4 Enable, VPN ID. This command works for physical interfaces and interfaces that are part of a port channel. It does not validate subinterfaces.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards

Command	Description	Supported Platforms
show consistency-checker l3 multicast group *ip-address* **source** *ip-address* **vrf** *vrf-name* [**brief** | **detail**]	Checks for inconsistencies with Layer 3 multicast groups.	Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-GX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards Cisco Nexus 9500 Series switches with N9K-X9432C-S, N9K-X9536PQ line cards; and N9K-X9432C-FM-S, N9K-C9508-FMX-S, and N9K-C9508-FM-S fabric modules. Cisco Nexus N3K-C3048TP, N3K-C3064-TC, N3K-C3232C, N3K-C3264Q, N3K-C31108TC-V, N3K-C3132Q-40GX, N3K-C3132Q-V, N3K-C31108PC-V, N3K-C3172PQ, N3K-C3172TQ, N3K-C3164Q, and N3K-C31128PQ-10GE switches. Cisco Nexus N9K-C9372TX, N9K-C9372TX-E, N9K-C93120TX, N9K-X9432C-S, N9K-C9332PQ, N9K-C9372PX and N9K-C9372PX-E switches.
show consistency-checker link-state fabric-ieth [**module** *module-number*] [**brief** | **detail**]	Verifies the programming consistency between software and hardware for the link-state status of internal fabric ports.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards
show consistency-checker link-state interface ethernet *slot/port* [**brief** | **detail**]	Verifies the programming consistency between software and hardware for the link-state status of the interfaces. This command works for physical Ethernet interfaces and physical Ethernet interfaces that are part of a port channel. It does not validate subinterfaces or FEX interfaces.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards

Command	Description	Supported Platforms
show consistency-checker link-state module *module-number* [**brief** | **detail**]	Verifies the software link state of all the interfaces in the module against its hardware link state. This command works for physical Ethernet interfaces and physical Ethernet interfaces that are part of a port channel. It does not validate subinterfaces or FEX interfaces.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards
show consistency-checker membership port-channels [**interface port-channel** *channel-number*] [**brief** | **detail**]	Checks for port-channel membership in the hardware in all modules and validates it with the software state. This command runs per port channel.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards
show consistency-checker membership port-channels [**brief** | **detail**]	Checks for port-channel membership in the hardware in all modules and validates it with the software state. This command runs for all port channels in the system.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards
show consistency-checker membership vlan *vlan-id* {**native-vlan** | **private-vlan interface** {**ethernet** *slot/port* | **port-channel** *number* | **native-vlan**}} [**brief** | **detail** | **interface**]	Determines that the VLAN membership in the software is the same as programmed in the hardware. It also ignores the interfaces that are in the STP BLK state.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards **Note** The **private-vlan** command does not support the **brief** or **detail** option. **Note** Cisco Nexus 34180YC platform switches support only the **native-vlan** command.
show consistency-checker pacl {**module** *module-number* | **port-channels interface port-channel** *channel-number*}	Validates the IPv4, IPv6, and MAC PACL programming consistency between the hardware and software and verifies if <label, entry-location> pairs are consistent between the hardware and software.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards
show consistency-checker pacl extended ingress {**ip** | **ipv6** | **mac**} **interface** {**ethernet** *slot/port* | **port-channel** *number*} [**brief** | **detail**]	Verifies PACL programming for ingress interfaces (including FEX interfaces) and port channels.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards

Command	Description	Supported Platforms
show consistency-checker pacl extended ingress {**ip** | **ipv6** | **mac**} **module** *module-number* [**brief** | **detail**]	Verifies PACL programming across all physical interfaces, subinterfaces, breakout ports, and FEX interfaces for the specified module.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards
show consistency-checker port-state fabric-ieth [**module** *module-number* [**ieth-port** *ieth-port*]] [**brief** | **detail**]	Verifies the state of internal fabric ports.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards
show consistency-checker port-state [**module** *module-number*] [**brief** | **detail**]	Verifies the port state for the specified module.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards
show consistency-checker racl {**module** *module-number* | **port-channels interface port-channel** *channel-number* | **svi interface vlan** *vlan-id*}	Validates the IPv4 and IPv6 RACL programming consistency between the hardware and software and verifies if <label, entry-location> pairs are consistent between the hardware and software. • When invoked per module, this command verifies IPv4 and IPv6 ACL consistency for all of the physical interfaces and subinterfaces for that module. • When invoked on a specific port channel, this command verifies for all the member ports. • When invoked on all port channels, this command verifies for each port channel that has an ACL applied. **Note** This command does not verify IPv4 and IPv6 ACLs and does not verify if qualifiers and actions are matching.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards
show consistency-checker racl extended ingress {**ip** | **ipv6**} **interface** {**ethernet** *slot/port* | **port-channel** *number* | **vlan** *vlan-id*} [**brief** | **detail**]	Verifies RACL programming for ingress interfaces, subinterfaces, breakout ports, port channels, or SVIs.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards

Command	Description	Supported Platforms
show consistency-checker racl extended ingress {**ip** | **ipv6**} **module** *module-number* [**brief** | **detail**]	Verifies RACL programming for ingress interfaces on the specified module. This command runs across all of the physical interfaces, subinterfaces, and breakout ports for that module.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards
show consistency-checker stp-state vlan *vlan-id* [**brief** | **detail** | **interface**]	Determines whether the spanning tree state in the software is the same as programmed in the hardware. This command is run only on interfaces that are operational (up).	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards.
show consistency-checker vacl extended ingress {**ip** | **ipv6** | **mac**} **vlan** *vlan-id* [**brief** | **detail**]	Verifies VACL programming on all of the member interfaces of the VLAN.	Cisco Nexus 34180YC, 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards
show consistency-checker vpc [**source-interface**] [**brief** | **detail**]	Checks for vPC inconsistencies. Check for LACP individual (I) state of the ports that does not have egress mask.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX and -FX line cards Cisco Nexus 9500 Series switches with N9K-X9432C-S, N9K-X9536PQ line cards; and N9K-X9432C-FM-S, N9K-C9508-FMX-S, and N9K-C9508-FM-S fabric modules. Cisco Nexus N3K-C3048TP, N3K-C3064-TC, N3K-C3232C, N3K-C3264Q, N3K-C31108TC-V, N3K-C3132Q-40GX, N3K-C3132Q-V, N3K-C31108PC-V, N3K-C3172PQ, N3K-C3172TQ, N3K-C3164Q, and N3K-C31128PQ-10GE switches. Cisco Nexus N9K-C9372TX, N9K-C9372TX-E, N9K-C93120TX, N9K-X9432C-S, N9K-C9332PQ, N9K-C9372PX and N9K-C9372PX-E switches.

Command	Description	Supported Platforms
show consistency-checker vxlan config-check [**verbose-mode**]	Verifies the VXLAN EVPN configuration on the switch.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches Cisco Nexus C31108PC-V, C31108TC-V, C3132Q-V and 3132C-Z switches. Cisco Nexus C9396TX, C93128TX, C9396PX, X9564PX, X9564TX and X9536PQ switches. Cisco Nexus C3132Q-40GE-SUP, C3132Q-40GX-SUP, C3132Q-XL, C31128PQ-10GE, C3264Q-S, C3264C-E switches.
show consistency-checker vxlan infra [**verbose-mode**]	Checks for inconsistencies with the VXLAN tunnel infrastructure.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches Cisco Nexus C31108PC-V, C31108TC-V, C3132Q-V and 3132C-Z switches. Cisco Nexus C9396TX, C93128TX, C9396PX, X9564PX, X9564TX and X9536PQ switches. Cisco Nexus C3132Q-40GE-SUP, C3132Q-40GX-SUP, C3132Q-XL, C31128PQ-10GE, C3264Q-S, C3264C-E switches.
show consistency-checker vxlan l2 module *module-number*	Verifies the consistency with VXLAN Layer 2 routes.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches Cisco Nexus C31108PC-V, C31108TC-V, C3132Q-V and 3132C-Z switches. Cisco Nexus C9396TX, C93128TX, C9396PX, X9564PX, X9564TX and X9536PQ switches. Cisco Nexus C3132Q-40GE-SUP, C3132Q-40GX-SUP, C3132Q-XL, C31128PQ-10GE, C3264Q-S, C3264C-E switches.

Command	Description	Supported Platforms
show consistency-checker vxlan l3 vrf [*vrf-name* | **all**] [**start-scan** | **report**]	Checks for inconsistencies with VXLAN Layer 3 routes.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches Cisco Nexus C31108PC-V, C31108TC-V, C3132Q-V and 3132C-Z switches. Cisco Nexus C9396TX, C93128TX, C9396PX, X9564PX, X9564TX and X9536PQ switches.
show consistency-checker vxlan pv	Verifies if VLAN mappings are programmed consistently between the software and across different tables in the hardware. At least one interface needs to be enabled with port VLAN mappings in order to run this command.	Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2, and 9500 platform switches Cisco Nexus C31108PC-V, C31108TC-V, C3132Q-V and 3132C-Z switches. Cisco Nexus C9396TX, C93128TX, C9396PX, X9564PX, X9564TX and X9536PQ switches.
show consistency-checker vxlan qinq-qinvni	Checks for a multi-tag VLAN list and associated multi-tag vn-segment being consistent in the software and hardware.	Cisco Nexus 9300-FX/FX2 platform switches Cisco Nexus C31108PC-V, C31108TC-V, C3132Q-V and 3132C-Z switches. Cisco Nexus C9396TX, C93128TX, C9396PX, X9564PX, X9564TX and X9536PQ switches.
show consistency-checker vxlan selective-qinvni interface {**ethernet** *slot/port* | **port-channel** *channel-number*}	Validates if port-specific selective Q-in-VNI mappings are programmed correctly in the software and hardware in order for the inner tags in the packets to be preserved.	Cisco Nexus 9300-EX and 9300-FX/FX2 platform switches Cisco Nexus C31108PC-V, C31108TC-V, C3132Q-V and 3132C-Z switches. Cisco Nexus C9396TX, C93128TX, C9396PX, X9564PX, X9564TX and X9536PQ switches.

Command	Description	Supported Platforms
show consistency-checker vxlan vlan [**all** | *vlan-id*] [**verbose-mode**]	Checks for inconsistencies with VXLAN VLANs.	Cisco Nexus 9300-EX and 9300-FX/FX2 platform switches Cisco Nexus C31108PC-V, C31108TC-V, C3132Q-V and 3132C-Z switches. Cisco Nexus C9396TX, C93128TX, C9396PX, X9564PX, X9564TX and X9536PQ switches. Cisco Nexus C3132Q-40GE-SUP, C3132Q-40GX-SUP, C3132Q-XL, C31128PQ-10GE, C3264Q-S, C3264C-E switches.
show consistency-checker tap-aggregation qinq	Checks for inconsistencies with port tap-aggregation and qinq.	Cisco Nexus N9K-C9316D-GX, N9K-C93600CD-GX, N9K-C9364C-GX, N9K-C9504-FM-G, and N9KC9508- FM-G switches and N9K-X9716D-GX line cards
show consistency-checker vxlan xconnect	Checks for inconsistencies with VXLAN Xconnect VLANs. Validates that Xconnect ACLs are installed on all units and slices and MAC learn is disabled on all Xconnect VLANs.	Cisco Nexus 9200, 9332C, 9364C, 9300-EX, and 9300-FX/FX2 platform switches
show consistency-checker vxlan l3 single-route [**ipv4** | **ipv6**] [*vrf*]	Checks for inconsistencies with VXLAN layer 3 single route traffic.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches. Cisco Nexus C31108PC-V, C31108TC-V, C3132Q-V and 3132C-Z switches. Cisco Nexus C9396TX, C93128TX, C9396PX, X9564PX, X9564TX and X9536PQ switches and Cisco Nexus 9200, 9300-EX and 9300-FX platform switches.

Command	Description	Supported Platforms
show consistency-checker vxlan l2 [**mac-address**] [*mac-address*] | **module**] [*module*	Checks for inconsistencies with VXLAN layer 2.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches. Cisco Nexus C31108PC-V, C31108TC-V, C3132Q-V and 3132C-Z switches. Cisco Nexus C9396TX, C93128TX, C9396PX, X9564PX, X9564TX and X9536PQ switches and Cisco Nexus 9200, 9300-EX and 9300-FX platform switches. Cisco Nexus C3132Q-40GE-SUP, C3132Q-40GX-SUP, C3132Q-XL, C31128PQ-10GE, C3264Q-S, C3264C-E switches.

Command	Description	Supported Platforms
show consistency-checker storm-control [**brief** | **detail**]	Checks for inconsistencies with storm control	Beginning with Cisco NX-OS Release 10.5(1), the breif and detail of the storm control consistency. Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, -FX, and -R line cards Beginning with Cisco NX-OS Release 9.3(5), it is supported on N3K-C3016Q-40GE,N3K-C3048TP-1GE, N3K-C3064PQ-10GE, N3K-C3064PQ-10GX, N3K-C3064T-10GT, N9K-C9504-FM, N9K-C9508-FM, N9K-C9516-FM, N9K-C9508-FM-S, N3K-C31128PQ, N3K-C3164Q-40GE, N3K-C3232C, N3K-C3132Q-V, N3K-C31108PC-V, N3K-C31108TC-V, N3K-C3264C-E, N3K-C3132C-Z, N9K-C93128TX, N9K-C9396PX, N9K-C9372PX and N9K-C9332PQ devices. **Note** When ND ISSU is done to Cisco NX-OS Release 10.4(x), and if pol_rate or pol_burst value in hardware and software do not match, the storm control consistency checker fails. To resolve the issue, reconfigure storm control.
show consistency-checker segment-routing mpls [**ip**] [*ip-address*] | **mask**] [*mask* | **vrf**] [*vrf*	Checks route consistency for Underlay Segment Routing (ISIS, BGP, OSPF) and Overlay routes Layer 3 VPN and Layer 2 EVPN.	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, and -FX line cards. Cisco Nexus N9K-C9316D-GX, N9K-C93600CD-GX, N9K-C9364C-GX devices.

Command	Description	Supported Platforms
show consistency-checker segment-routing mpls label	Checks label consistency for Underlay Segment Routing (ISIS, BGP, OSPF) and Overlay routes Layer 3 VPN, Layer 2 EVPN, and ADJ SIDS	Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches and Cisco Nexus 9500 platform switches with -EX, and -FX line cards. Cisco Nexus N9K-C9316D-GX, N9K-C93600CD-GX, N9K-C9364C-GX devices.
show consistency-checker sflow [**brief** | **detail**]	Checks the program and consistency configurations for supervisor and line cards hardware tables.	Cisco Nexus 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 platform switches **Note** Beginning with Cisco NX-OS Release 10.3(3)F, Cisco Nexus 9808 platform switches

The following commands do not support JSON output:

- **show consistency-checker forwarding** {**ipv4** | **ipv6**} [**vrf** *vrf-name* | **all**] [**module** *module-number* | **all**]
- **show consistency-checker pacl** {**module** *module-number* | **port-channels interface port-channel** *channel-number*}
- **show consistency-checker racl module** *module-number*
- **show consistency-checker racl port-channels interface port-channel** *channel-number*}
- **show consistency-checker racl svi interface vlan** *vlan-id*
- **show consistency-checker vxlan**
- **test consistency-checker forwarding** {**ipv4** | **ipv6**} [**vrf** *vrf-name* | **all**] [**module** *module-number* | **all**]

The **show consistency-checker vxlan** commands are not modeled.

Multicast Consistency Checker

The multicast consistency checker is a single-route consistency checker for Layer 2 and Layer 3 routes for verifying the state of multicast routes. The multicast consistency checker executes the show commands in each component, parses the relevant information, and then compares the processed information against the other components to check for inconsistencies. The multicast consistency checker commands terminate upon encountering a failure. The **show consistency-checker l2 multicast group** and **show consistency-checker l3 multicast group** commands return the differences in the expected value and the actual value.

The commands support the following output formats:

- **verbose**: Displays the results in text format.
- **detail**: Displays the results in JSON format.
- **brief**: Displays the results in JSON format with minimal details.

Beginning with Cisco NX-OS Release 10.2(2)F, L3 Multicast Consistency Checker supports NAT translation and is supported on all platforms. UMNAT is not supported.

Note MMNAT stands for Multicast to Multicast NAT, MUNAT stands for Multicast to Unicast NAT, and UMNAT stands for Unicast to Multicast NAT. NAT translation must be of the type MMNAT ingress and egress, and MUNAT.

Beginning with Cisco NX-OS Release 10.2(1)F, Multicast over GRE consistency checker is introduced on N9K-C9316D-GX, N9K-C93600CD-GX, N9K-C9364C-GX family switches. Multicast over GRE (mGRE) Consistency Checker supports the following:

- Single route mGRE Consistency Checker
- mGRE tunnels on L3 Ethernet Interfaces, L3 Port-channels and L3 sub-interfaces
- GRE tunnel where transport protocol VRF can be different from tunnel interface VRF. This is supported only for GREv4 - GRE tunnels over IPv4 multicast.

Multicast over GRE (mGRE) Consistency Checker does not support the following:

- FEX
- GRE tunnels over IPv6
- mGRE is not supported on EoRs. Consistency check is supported only on N9K-C9316D-GX, N9KC93600CD-GX, N9K-C9364C-GX ToRs.
- mGRE is not supported on SVIs.

The mGRE consistency checks happen only if there is a IP GRE Tunnel interface in the outgoing interface list or if the RPF interface is an IP GRE tunnel interface.

Beginning from Cisco NX-OS Release 10.1(1), the following consistency checkers are supported:

- IPv6 L2 Multicast Consistency Checker
- IPv6 L3 Multicast Consistency Checker
- Multicast NLB Consistency Checker
  - Multicast MAC Lookup mode Consistency Checker
  - Multicast NLB L3 unicast configuration Consistency Checker
- Multicast GRE Consistency Checker

The following existing CLI command is extended to accept IPv6 source and group addresses for IPv6 L2 Multicast Consistency Checker:

show consistency-checker l2 multicast group <ipv4/ipv6 group address> source <ipv4/v6 source address> vrf <vrf-id> [brief|detail]

The following is the output example for IPv6 L2 Multicast Consistency Checker:

```
# show consistency-checker l2 multicast group ?
  A.B.C.D   Group IP address
  A:B::C:D  Group IPv6 address
```

The following existing CLI command is extended to accept IPv6 source and group addresses for IPv6 L3 Multicast Consistency Checker:

show consistency-checker l3 multicast group <ipv4/ipv6 group address> source <ipv4/v6 source address> vlan <vlan-id> [brief|detail]

The following is the output example for IPv6 L3 Multicast Consistency Checker:

```
# show consistency-checker l3 multicast group ?
  A.B.C.D   Group IP address
  A:B::C:D  Group IPv6 address
```

The following new CLI command is added to support Multicast MAC Lookup mode Consistency Checker:

show consistency-checker l2 multicast mac <mac> vlan <vlan-id>

The following is the output example for Multicast MAC Lookup mode Consistency Checker:

```
# show consistency-checker l2 multicast mac 0100.1234.1234 vlan 10 ?
  >       Redirect it to a file
  >>      Redirect it to a file in append mode
  brief   Show consistency checker structured output in brief
  detail  Show consistency checker structured output in detail
  |       Pipe command output to filter
```

Note This CLI is used for MAC lookup mode Consistency Checker or L2 mode consistency checker for NLB. The input MAC can be ip-mac or non-ip-mac.

The following new CLI command is added to support Multicast NLB L3 unicast configuration Consistency Checker:

show consistency-checker multicast nlb cluster-ip <unicast-cluster-ip> vrf <vrf-id>

The following is the output example for Multicast NLB L3 unicast configuration Consistency Checker:

```
# show consistency-checker multicast nlb cluster-ip <unicast-cluster-ip>
  >       Redirect it to a file
  >>      Redirect it to a file in append mode
  brief   Show consistency checker structured output in brief
  detail  Show consistency checker structured output in detail
  |       Pipe command output to filter
```

The following existing CLI command is used for Multicast GRE Consistency Checker:

show consistency-checker l3 multicast group <ipv4 group address> source <ipv4 source address> vrf <vrf-id> [brief|detail]

Note Existing IPv4 L3 multicast consistency checker CLI will be used to start Multicast GRE Consistency Checker.

The multicast consistency checker supports the following devices:

- Cisco Nexus 92304QC, 9272Q, 9232C, 9236C, 92300YC, 93108TC-EX, 93180LC-EX, 93180YC-EX, and 9300-GX platform switches and N9K-X9736C-EX, N9K-X97160YC-EX, N9K-X9732C-EX, and N9K-X9732C-EXM line cards.
- Cisco Nexus 9500 Series switches with N9K-X96136YC-R, N9K-X9636C-R, and N9K-X9636Q-R line cards.

Beginning with Cisco NX-OS Release 9.3(5), multicast consistency checker supports the following devices:

- Cisco Nexus 9500 Series switches with N9K-X9432C-S, N9K-X9536PQ line cards; and N9K-X9432C-FM-S, N9K-C9508-FMX-S, and N9K-C9508-FM-S fabric modules.
- Cisco Nexus N3K-C3048TP, N3K-C3064-TC, N3K-C3232C, N3K-C3264Q, N3K-C31108TC-V, N3K-C3132Q-40GX, N3K-C3132Q-V, N3K-C31108PC-V, N3K-C3172PQ, N3K-C3172TQ, N3K-C3164Q, and N3K-C31128PQ-10GE switches.
- Cisco Nexus N9K-C9372TX, N9K-C9372TX-E, N9K-C93120TX, N9K-X9432C-S, N9K-C9332PQ, N9K-C9372PX, and N9K-C9372PX-E switches.

Beginning with Cisco NX-OS Release 10.1(1), multicast consistency checker supports the following devices:

- Cisco Nexus N9k-C9504 with N9K-X97160YC-EX, N9k-C9504 with N9K-X9732C-EX, N9k-C9504 with N9K-X9732C-FX, N9k-C9504 with N9K-X9736C-EX, N9k-C9504 with N9K-X9736C-FX, N9k-C9504 with N9K-X9736Q-FX, and N9k-C9504 with N9K-X9788TC-FX.
- Cisco Nexus N9k-C9508 with N9K-X97160YC-EX, N9k-C9508 with N9K-X9732C-EX, N9k-C9508 with N9K-X9732C-FX, N9k-C9508 with N9K-X9736C-EX, N9k-C9508 with N9K-X9736C-FX, N9k-C9508 with N9K-X9736Q-FX, and N9k-C9508 with N9K-X9788TC-FX.
- Beginning with Cisco NX-OS Release 10.3(1)F, Multicast Consistency Checker is supported on the Cisco Nexus 9808 platform switches.
    - Beginning with Cisco NX-OS Release 10.4(1)F, Multicast Consistency Checker is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.

Beginning with Cisco NX-OS Release 10.4(1)F, Multicast Consistency Checker is supported on the Cisco Nexus 9804 platform switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.

Beginning with Cisco NX-OS Release 10.4(2)F, Multicast Consistency Checker is supported on the Cisco Nexus 9232E-B1 platform switches.

The Multicast Consistency Checker verifies the programming consistency of the following Layer 2 components:

- IGMP snooping
- MFDM
- MFIBPI
- MFIBPD
- Hardware tables

The Multicast Consistency Checker verifies the programming consistency of the following Layer 3 components:

- PIM
- MRIB
- IGMP snooping
- MFDM
- MFIBPI
- MFIBPD

- Hardware tables

Beginning with Cisco NX-OS Release 10.5(3)F, Layer 3 Consistency Checker is supported on the Cisco Nexus N9364E-SG2-Q platform switches.

Output Examples for Multicast Consistency Checker Commands

The following is an example of IGMP snooping output:

```
switch# show ip igmp snooping groups 225.12.12.28 225.12.12.28 vlan 222
Type: S - Static, D - Dynamic, R - Router port, F - Fabricpath core port
Vlan  Group Address      Ver  Type  Port list
222   225.12.12.28       v3   D     Eth1/2 Eth1/3 Po12 Po100 Po18
```

The following is an example of MFDM output:

```
switch# show forwarding distribution l2 multicast vlan 222 group 225.12.12.28 source
225.12.12.28
Vlan: 222, Group: 225.12.12.28, Source: 225.12.12.28
  Outgoing Interface List Index: 4
  Reference Count: 204
  Num L3 usages: 4
  Platform Index: 0xa00004
  Vpc peer link exclude flag set
  Number of Outgoing Interfaces: 5
    Ethernet1/2
    Ethernet1/3
    port-channel12
    port-channel18
    port-channel100
```

The following is an example of comparing IGMP Snooping with MFDM (passed):

```
**********************************************************************
Comparing IGMP Snooping with MFDM
**********************************************************************
L2 Eth Receivers :
IGMP Snooping: 1/2, 1/3
MFDM: 1/2, 1/3

L2 PC Receivers :
IGMP Snooping: 100, 12, 18
MFDM: 12, 100, 18

CC between IGMP Snooping and MFDM PASSED
```

The following is an example of comparing IGMP Snooping with MFDM (failed):

```
**********************************************************************
Comparing IGMP Snooping with MFDM
**********************************************************************
L2 Eth Receivers:
IGMP Snooping: 1/2, 1/3
MFDM: 1/2,  1/3

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
L2 PC Receivers:
```

```
IGMP Snooping: 100, 12, 18
MFDM: 12, 100, 16
Consistency check failed!!!
Missing elements are: 18
Additional elements are: 16
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Congestion Detection and Avoidance

Beginning Cisco NX-OS Release 9.3(3), Cisco Nexus 9000 series switches supports **show tech-support slowdrain** command to troubleshoot congestion issues. The **show tech-support slowdrain** command contains some of the congestion detection indications, counters, and log messages as well as other commands that allow an understanding of the switches, Cisco NX-OS versions, and topology.

Since, congestion can propagate from one switch to another, you must gather the **show tech-support slowdrain** command output from all the switches at the same time for a better assessment of the congestion triggers and propogation.

Fan and Thermal Monitoring Commands

Cisco NX-OS introduces enhanced environmental monitoring and fan speed management capabilities. The commands provide real-time insights into optical module temperatures, fan speed changes, and related system events. These tools are vital for maintaining optimal switch operation, diagnosing overheating, and understanding automated fan control behavior.

Supported Platforms

These commands are supported on following Cisco Nexus switches :

- N9364E-SG2-Q
- N9364E-SG2-O

Fan and Thermal Monitoring Commands

Table 5: Fan and Thermal Monitoring Commands

Command	Description	Supported Platforms
show env temperature transceiver	Displays current temperature, warning, and alarm thresholds for all optical transceivers. Indicates status as OK or ALARM per interface.	Cisco Nexus N9364E-SG2-Q, N9364E-SG2-O
show logging | include fan|temp|alarm	Displays system log messages related to temperature alarms and fan speed changes for historical event tracking.	Cisco Nexus N9364E-SG2-Q, N9364E-SG2-O

Command Output Examples and Field Descriptions

The following examples demonstrate output and explain key fields for each command:

show env temperature transceiver

```
interface      temperature   high-warning   high-temp-warn-thr   status
----------------------------------------------------------------------
Ethernet1/17   19           75             72                   OK
Ethernet1/34   44           70             67                   OK
Ethernet1/35   68           70             67                   ALARM
```

Field Descriptions:

- **interface** – Interface identifier for the optic module.
- **temperature** – Current measured temperature (°C).
- **high-warning** – Manufacturer warning threshold (°C).
- **high-temp-warn-thr** – Platform-defined alarm threshold (°C) that may trigger fan speed increase.
- **status** – OK if within limits, ALARM if threshold exceeded.

Note

- Values update every minute.
- An ALARM status indicates optics overheating and fan speed escalation.

show logging | include fan|temp|alarm

```
2025 Aug 07 10:01:06 %USER-3-SYSTEM_MSG: Temperature alarm triggered on interface Ethernet1/34
 - pfm
2025 Aug 07 10:02:00 %USER-3-SYSTEM_MSG: Temperature alarm cleared on interface Ethernet1/34
 - pfm
```

Field Descriptions:

- **Temperature alarm triggered** – Sensor threshold exceeded.
- **Temperature alarm cleared** – Sensor value normal.

Best Practices and Usage Tips

- Regularly monitor optical temperatures and status using **show env temperature transceiver** for early detection of overheating.
- Correlate ALARM status with **show logging** output to validate automated fan response.
- Allow up to two minutes for new sensor readings to propagate due to polling intervals.
- Use **show logging** for historical analysis of temperature when troubleshooting.

ACL Consistency Checker

Beginning with Cisco NX-OS Release 9.3(3), the ACL consistency checker supports the following devices:

N9K-C9372PX, N9K-C9372PX-E, N9K-C9372TX, N9K-C9372TX-E, N9K-C9332PQ, N9K-C93128TX, N9K-C9396PX, N9K-C9396TX, N9K-C9508-FM-S, N9K-C9508-FM2, N9K-C9504-FM-S, N9K-X9632PC-QSFP100, N9K-X9432C-S

Beginning with Cisco NX-OS Release 9.3(5), the ACL consistency checker is supported on Cisco Nexus N9K-C9316D-GX, N9K-C93600CD-GX, N9K-C9364C-GX, N9K-C93240YC-FX2, N9K-C93180YC-EX, N3K-C3636C-R, N3K-C36180YC-R; and Cisco Nexus 9500 Series switches with N9K-X9636Q-R, N9K-X9636C-R, N9K-X9636C-RX and N9K-X96136YC-R line cards.

Beginning with Cisco NX-OS Release 10.3(1)F, ACL Consistency Checker is supported on the Cisco Nexus 9808 platform switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, ACL Consistency Checker is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.

Beginning with Cisco NX-OS Release 10.4(1)F, ACL Consistency Checker is supported on the Cisco Nexus 9804 platform switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.

The following entities are verified as part of the ACL consistency check:

Action, Protocol, SIP, DIP, source port, destination port, Source MAC, Destination MAC, Ethertype, COS, DSCP, VLAN and UDFs.

Cisco NX-OS supports the following PACL, RACL, and VACL consistency checker commands.

Command	Description
show consistency-checker pacl extended ingress ip module <module-id> [brief | detail]	Verifies PACL consistency check for ingress interfaces and port channel for the specified IP module.
show consistency-checker pacl extended ingress ipv6 module <module-id> [brief | detail]	Verifies PACL consistency check for ingress interfaces and port channel for the specified IPv6 module.
show consistency-checker pacl extended ingress mac module <module-id> [brief | detail]	Verifies MAC PACL consistency check for ingress interfaces and port channel for the specified MAC module.
show consistency-checker pacl extended ingress ip interface {<int-id> | <ch-id> [brief | detail]	Verifies PACL consistency check for the specified ingress interface.
show consistency-checker pacl extended ingress ipv6 interface {<int-id> | <ch-id> [brief | detail]	Verifies PACL consistency check for the specified IPv6 ingress interface.
show consistency-checker pacl extended ingress mac interface {<int-id> | <ch-id> [brief | detail]	Verifies PACL consistency check for the specified ingress MAC interface.
show consistency-checker racl extended ingress ip module <module-id> [brief | detail]	Verifies RACL consistency check for ingress interfaces and port channel for the specified IP module.
show consistency-checker racl extended ingress ipv6 module <module-id> [brief | detail]	Verifies RACL consistency check for ingress interfaces and port channel for the specified IPv6 module.

Command	Description
show consistency-checker racl extended ingress ip interface {<int-id> | <ch-id> | <vlan-id>} [brief | detail]	Verifies RACL consistency check for the specified ingress interface.
show consistency-checker racl extended egress ip interface {<int-id> | <ch-id> | <vlan-id>} [brief | detail]	Verifies RACL consistency check for the specified egress IP interface.
show consistency-checker racl extended ingress ipv6 interface {<int-id> | <ch-id> | <vlan-id>} [brief | detail]	Verifies RACL consistency check for the specified ingress IPv6 interface.
show consistency-checker racl extended egress ipv6 interface {<int-id> | <ch-id> | <vlan-id>}[brief | detail]	Verifies RACL consistency check for the specified egress IPv6 interface.
show consistency-checker vacl extended ingress ip vlan <vlan-id> [brief | detail]	Verifies VACL consistency check for the specified IP VLAN.
show consistency-checker vacl extended ingress ipv6 vlan <vlan-id> [brief | detail]	Verifies VACL consistency check for the specified IPv6 VLAN.
show consistency-checker vacl extended ingress mac vlan <vlan-id> [brief | detail]	Verifies VACL consistency check for the specified ingress MAC VLAN.

Output Examples for ACL Consistency Checker Commands

This example shows the RACL consistency check results:

```
switch# show consistency-checker racl extended ingress ip module 1 Consistency checker
passed for Eth1/3 (ingress, ip, ip-list)
switch#
switch#
switch# show consistency-checker racl extended ingress ip module 1 brief
{
   "result": {
   "status": "CC_STATUS_OK",
   "checkers": [
     {
       "version": 1,
       "type": "CC_TYPE_IF_RACL",
       "status": "CC_STATUS_OK",
        "platformDetails": {
        "classType": "CC_PLTFM_NXOS_BCM"
      },
      "recoveryActions": [],
      "failedEntities": []
      }
    ]
   }
}
switch#
switch # show consistency-checker racl extended ingress ip interface ethernet 3/5
Consistency checker passed for Ethernet3/5 (ingress, ip, ip-list)
switch#
switch# show consistency-checker racl extended ingress ip interface ethernet 3/5 brief
{
   "result": {
```

```
   "status": "CC_STATUS_OK",
    "checkers": [
      {
       "version": 1,
       "type": "CC_TYPE_IF_RACL",
       "status": "CC_STATUS_OK",
       "platformDetails": {
       "classType": "CC_PLTFM_NXOS_BCM"
      },
       "recoveryActions": [],
       "failedEntities": []
     }
    ]
  }
}
```

Proactive Consistency Checker

Consistency check between software and hardware tables on Nexus platform is a high priority serviceability challenge with respect to route consistency checker. The existing route consistency checker is not a proactive mechanism and is an on-demand consistency checker when a command is issued.

The Proactive Consistency checker has a route/adjacency consistency checker that runs in the background continuously that enables to pro-actively detect any inconsistency for IPv4 or IPv6 routes and ARP or ND adjacencies.

Beginning with Cisco NX-OS Release 10.3(1)F, Proactive Consistency Checker is supported on Cisco Nexus 9504/9508 modular chassis with R/RX line cards.

The Proactive Consistency Checker is supported on all Cloudscale EOR and TOR platforms. It has two types of consistency checking methods.

- **The Full Database Consistency Checker**: This performs the consistency check on complete route and adjacency database.
- **The Incremental Consistency Checker**: This consistency check runs on the incremental change set of routes and adjacencies which got updated or added over a period of time.

Beginning with Cisco NX-OS Release 10.3(2)F, Proactive consistency checker will be supporting MPLS route consistency check for IPv4, IPv6, VPNv4, VPNv6, and PE/Deagg FEC types on Cisco Nexus 9504 and 9508 modular chassis with R/R2/RX line cards.

Show commands

Whenever any inconsistency is found by the proactive consistency checker, the following syslog will be generated:

"%UFDM-3-PROACTIVE_CC_INCONSISTENCY_FOUND: Inconsistencies found in Proactive CC session"

The following two commands must be used to check the inconsistencies during proactive consistency check:

Commands	Description
show forwarding proactive-cc inconsistencies	This show command displays the inconsistencies found in the last failed iteration.

Commands	Description
show forwarding proactive-cc inconsistencies all	This show command displays all the inconsistencies found from the time when the proactive consistency check is configured

If the user intends to clear the inconsistencies seen in the above two commands, the following command can be used:

"clear forwarding proactive-cc inconsistencies"

Configuration Commands

The following are the commands to enable/disable the feature and to change the periodicity (timer) for incremental and full consistency check:

- **platform proactive-cc forwarding** (enables with default timers)
- **no platform proactive-cc forwarding** (to disable)
- **platform proactive-cc forwarding fulldb <time in sec>**
- **platform proactive-cc forwarding incremental <time in sec>**
- **platform proactive-cc forwarding incremental <time in sec> fulldb <time in sec>**

Command	Purpose
platform proactive-cc forwarding **Example:** `switch(config)# platform proactive-cc forwarding`	This command enables the proactive consistency checker in the switch and default timers will be set. FulldB default timer value is 86400. Incremental dB default timer value is 10 seconds.
no platform proactive-cc forwarding **Example:** `switch(config)# no platform proactive-cc forwarding`	This command disables the proactive consistency checker.
platform proactive-cc forwarding fulldb <time in sec> **Example:** `switch(config)# platform proactive-cc forwarding fulldb 600`	This command will configure proactive consistency checker fulldB timer to 600 seconds.
platform proactive-cc forwarding incremental <time in sec> **Example:** `switch(config)# platform proactive-cc forwarding incremental 20`	This command will configure proactive cc incremental timer value to 20 seconds.

Command	Purpose
platform proactive-cc forwarding incremental <time in sec> fulldb <time in sec> **Example:** `switch(config)# platform proactive-cc forwarding incremental 20 fulldb 600`	This command will configure both incremental timer and fulldB timer together.

Interface Consistency Checker

Beginning with Cisco NX-OS Release 10.3(1)F, Interface Consistency Checker is supported on the Cisco Nexus 9808 platform switches.

Beginning with Cisco NX-OS Release 10.4(1)F, Interface Consistency Checker is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.

Beginning with Cisco NX-OS Release 10.4(2)F, Interface Consistency Checker is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9232E-B1 switches.

Beginning with Cisco NX-OS Release 10.4(1)F, Interface Consistency Checker is supported on the Cisco Nexus 9804 platform switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.

ITD Consistency Checker

ITD generates settings on dependent components internally to achieve expected functionality. Any unexpected setting on these components results in an ITD malfunction. The ITD consistency-checker through CLI, displays if any inconsistency is found between the ITD and actual settings on these components.

ITD consistency-check is stop-on-error, which means if a property check fails for service, ITD skips checking the remaining properties and replies with a failure for that service.

For example: When running the **show consistency-checker itd all** [**brief** | **detail**] command, if one property check fails for one service, ITD will move on to check the next service.

Beginning with Cisco NX-OS Release 10.3(2)F, the following ITD consistency checker commands are supported on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 platform switches:

Command	Description
show consistency-checker itd <*service-name*> [**brief** | **detail**]	Displays the consistency-check on one service <*service-name*>. If the service does not exist, the check will be skipped.
show consistency-checker itd all [**brief** | **detail**]	Displays the consistency-check for each existing ITD service in order and responses with the result if check for each service is passed or failed.

Command	Description
show consistency-checker itd ingress interface < *intf-name*> **source** <*srcIP*> **destination** <*destIP*> [**brief** | **detail**]	Displays whether the ITD service consistency-checker is passed or failed if the given flow to the ingress interface hits a redirect policy generated by an ITD service. If the flow is not hitting any ITD generated policy, the service consistency-check will treat as passed.

Configuration Files

Configuration files contain the Cisco NX-OS commands used to configure the features on a Cisco NX-OS device. Cisco NX-OS has two types of configuration files: running configuration and startup configuration. The device uses the startup configuration (startup-config) during the device startup to configure the software features. The running configuration (running-config) contains the current changes that you make to the startup-configuration file. You should create a backup version of your configuration files before modifying that configuration. You can back up the configuration files to a remote server. See the configuration file information in the *Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*. You can also create a checkpoint copy of the configuration file that you can roll back to if problems occur. See the rollback feature in the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

Cisco NX-OS features can create internal locks on the startup configuration file. In rare cases, these locks might not be removed by the features. Use the **show system internal sysmgr startup-config locks** command to determine if any locks remain on the startup configuration file. Use the **system startup-config unlock** command to remove these locks.

CLI Debug

Cisco NX-OS supports an extensive debugging feature set for actively troubleshooting a network. Using the CLI, you can enable debugging modes for each feature and view a real-time updated activity log of the control protocol exchanges. Each log entry has a time stamp and is listed chronologically. You can limit access to the debug feature through the CLI roles mechanism to partition access on a per-role basis. While the **debug** commands show real-time information, you can use the **show** commands to list historical and real-time information.

Caution Use the **debug** commands only under the guidance of your Cisco technical support representative because some **debug** commands can impact your network performance.

Note You can log debug messages to a special log file, which is more secure and easier to process than sending the debug output to the console.

By using the **?** option, you can see the options that are available for any feature. A log entry is created for each entered command in addition to the actual debug output. The debug output shows a time-stamped account of the activity that occurred between the local device and other adjacent devices.

You can use the debug facility to track events, internal messages, and protocol errors. However, you should be careful when using the debug utility in a production environment because some options might prevent access to the device by generating too many messages to the console or creating CPU-intensive events that could seriously affect network performance.

Note We recommend that you open a second Telnet or SSH session before you enter any **debug** commands. If the debug session overwhelms the current output window, you can use the second session to enter the **undebug all** command to stop the debug message output.

Debug Filters

You can filter out unwanted debug information by using the **debug-filter** command. The **debug-filter** command allows you to limit the debug information produced by related **debug** commands.

The following example limits EIGRP hello packet debug information to Ethernet interface 2/1:

```
switch# debug-filter ip eigrp interface ethernet 2/1
switch# debug eigrp packets hello
```

Ping, Pong, and Traceroute

Note Use the ping and traceroute features to troubleshoot problems with connectivity and path choices. Do not use these features to identify or resolve network performance issues. Use the pong feature to measure the delay of the network between two points.

The **ping** and **traceroute** commands are two of the most useful tools for troubleshooting TCP/IP networking problems. 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.

The traceroute utility operates in a similar fashion but can also determine the specific path that a frame takes to its destination on a hop-by-hop basis.

The **pong** utility can measure the delay of the network between two points.

Using Ping

Use the **ping** command to verify connectivity and latency to a particular destination across an IPv4 routed network.

Use the **ping6** command to verify connectivity and latency to a particular destination across an IPv6 routed network.

The ping utility allows you to send a short message to a port or end device. By specifying the IPv4 or IPv6 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.

Note We do not recommend using the Ping utility to test network performance with the IP address configured on the system.

Note We do not recommend using the Ping utility to test network performance with the IP address configured on the Nexus switch. ICMP (Ping) traffic directed to the switch IP address is subject to CoPP (Control Plane Policing) and may be dropped.

```
switch# ping 172.28.230.1 vrf management
PING 172.28.230.1 (172.28.230.1): 56 data bytes
64 bytes from 172.28.230.1: icmp_seq=0 ttl=254 time=1.095 ms
64 bytes from 172.28.230.1: icmp_seq=1 ttl=254 time=1.083 ms
64 bytes from 172.28.230.1: icmp_seq=2 ttl=254 time=1.101 ms
64 bytes from 172.28.230.1: icmp_seq=3 ttl=254 time=1.093 ms
64 bytes from 172.28.230.1: icmp_seq=4 ttl=254 time=1.237 ms

--- 172.28.230.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 1.083/1.121/1.237 ms
```

Using Traceroute

Use traceroute to do the following:

- Trace the route followed by the data traffic.
- Compute the interswitch (hop-to-hop) latency.

The traceroute utility 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 device and the device closest to the destination.

Use the **traceroute** {*dest-ipv4-addr* | *hostname*} [**vrf** *vrf-name*] command for IPv4 networks and the **traceroute6** {*dest-ipv6-addr* | *hostname*} [**vrf** *vrf-name*] command for IPv6 networks. If the destination cannot be reached, the path discovery traces the path up to the point of failure.

```
switch# traceroute 172.28.254.254 vrf management
traceroute to 172.28.254.254 (172.28.254.254), 30 hops max, 40 byte packets
 1  172.28.230.1 (172.28.230.1)  0.941 ms  0.676 ms  0.585 ms
 2  172.24.114.213 (172.24.114.213)  0.733 ms  0.7 ms  0.69 ms
 3  172.20.147.46 (172.20.147.46)  0.671 ms  0.619 ms  0.615 ms
 4  172.28.254.254 (172.28.254.254)  0.613 ms  0.628 ms  0.61 ms
```

Press **Ctrl-C** to terminate a running traceroute.

You can use the following commands to specify a source interface for the traceroute:

Command	Purpose
traceroute {*dest-ipv4-addr* | *hostname*} [**source** {*dest-ipv4-addr* | *hostname* | *interface*}] [**vrf** *vrf-name*] **Example:** `switch# traceroute 112.112.112.1 source vlan 10`	Specifies the source IPv4 address of the traceroute packets from the specified IP address, hostname, or interface.
traceroute6 {*dest-ipv6-addr* | *hostname*} [**source** {*dest-ipv6-addr* | *hostname* | *interface*}] [**vrf** *vrf-name*] **Example:** `switch# traceroute6 2010:11:22:0:1000::1 source ethernet 2/2`	Specifies the source IPv6 address of the traceroute6 packets from the specified IP address, hostname, or interface.
[**no**] **ip traceroute source-interface** *interface* [**vrf** *vrf-name*] **Example:** `switch(config)# ip traceroute source-interface loopback 1`	Generates traceroute or traceroute6 packets with the source IP address from the configured interface.

show ip traceroute source-interface [**vrf** *vrf-name*]
Example:
`switch# show ip traceroute source-interface vrf all`
`VRF Name Interface`
`default loopback1`
	Displays the configured source interface for the traceroute.

ip icmp-errors source-interface *interface*
Example 1:
`switch(config)# ip icmp-errors source-interface loopback 1`
Example 2:
`switch(config)# vrf context vrf-blue`
`switch(config-vrf)# ip icmp-errors source-interface loopback 2`
	Generates ICMP error packets with the source IPv4 or IPv6 address from the configured interface.
	You can also optionally configure this command within a virtual routing and forwarding instance (VRF).

Monitoring Processes and CPUs

Use the **show processes** command to identify the processes that are running and the status of each process. The command output includes the following:

- PID = process ID.

- State = process state.
- PC = current program counter in hexadecimal 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 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 Typically, the ER state designates a process that has been restarted too many times, causing the system to classify it as faulty and disable it.

```
switch# show processes ?
cpu       Show processes CPU Info
log       Show information about process logs
memory    Show processes Memory Info
```

```
switch# show processes
PID    State  PC        Start_cnt    TTY   Type  Process
-----  -----  --------  -----------  ----  ----  -------------
    1      S  b7f9e468            1     -     O  init
    2      S         0            1     -     O  migration/0
    3      S         0            1     -     O  ksoftirqd/0
    4      S         0            1     -     O  desched/0
    5      S         0            1     -     O  migration/1
    6      S         0            1     -     O  ksoftirqd/1
    7      S         0            1     -     O  desched/1
    8      S         0            1     -     O  events/0
    9      S         0            1     -     O  events/1
   10      S         0            1     -     O  khelper
   15      S         0            1     -     O  kthread
   24      S         0            1     -     O  kacpid
  103      S         0            1     -     O  kblockd/0
  104      S         0            1     -     O  kblockd/1
  117      S         0            1     -     O  khubd
  184      S         0            1     -     O  pdflush
  185      S         0            1     -     O  pdflush
  187      S         0            1     -     O  aio/0
  188      S         0            1     -     O  aio/1
  189      S         0            1     -     O  SerrLogKthread
```

```
...
```

Using the show processes cpu Command

Use the **show processes cpu** command to display CPU utilization. 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 = Average CPU time, in microseconds, for each process invocation.
- 1Sec = Percentage of CPU utilization for the last 1 second.

```
switch# show processes cpu
PID    Runtime(ms)  Invoked   uSecs  1Sec   Process
-----  -----------  --------  -----  -----  -----------
    1         2264    108252     20      0  init
    2          950    211341      4      0  migration/0
    3         1154  32833341      0      0  ksoftirqd/0
    4          609    419568      1      0  desched/0
    5          758    214253      3      0  migration/1
    6         2462  155309355     0      0  ksoftirqd/1
    7         2496    392083      6      0  desched/1
    8          443    282990      1      0  events/0
    9          578    260184      2      0  events/1
   10           56      2681     21      0  khelper
   15            0        30     25      0  kthread
   24            0         2      5      0  kacpid
  103           81        89    914      0  kblockd/0
  104           56       265    213      0  kblockd/1
  117            0         5     17      0  khubd
  184            0         3      3      0  pdflush
  185         1796    104798     17      0  pdflush
  187            0         2      3      0  aio/0
  188            0         2      3      0  aio/1
  189            0         1      3      0  SerrLogKthread
...
```

Using the show system resources Command

Use the **show system resources** command to display system-related CPU and memory statistics. The output includes the following:

- Load average 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 are actually running when the command is issued.
- CPU states show the CPU usage percentage in user mode, kernel mode, and idle time in the last 1 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.

```
switch# show system resources
Load average:   1 minute: 0.00   5 minutes: 0.02   15 minutes: 0.05
Processes   :   355 total, 1 running
```

```
CPU states  :   0.0% user,   0.2% kernel,   99.8% idle
        CPU0 states  :   0.0% user,   1.0% kernel,   99.0% idle
        CPU1 states  :   0.0% user,   0.0% kernel,   100.0% idle
        CPU2 states  :   0.0% user,   0.0% kernel,   100.0% idle
        CPU3 states  :   0.0% user,   0.0% kernel,   100.0% idle
Memory usage:   16402560K total,   2664308K used,   13738252K free
Current memory status: OK
```

Using Onboard Failure Logging

Cisco NX-OS provides the facility to log failure data to the persistent storage, which can be retrieved and displayed for analysis. This onboard failure logging (OBFL) feature stores failure and environmental information in nonvolatile memory on the module. This information will help you analyze failed modules.

The data stored by the OBFL facility includes the following:

- Time of initial power on
- Slot number of the module in the chassis
- Initial temperature of the module
- Firmware, BIOS, FPGA, and ASIC versions
- Serial number of the module
- Stack trace for crashes
- CPU hog information
- Memory leak information
- Software error messages
- Hardware exception logs
- Environmental history
- OBFL specific history information
- ASIC interrupt and error statistics history
- ASIC register dumps

For more information about configuring OBFL, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

Using OBFL Error Status Command

Beginning Cisco NX-OS Release 9.3(3), Cisco Nexus 9000 series switches supports several counters to monitor and log fibre channel interfaces. The counters help identify and troubleshoot issues at FCMAC level.

Use the **show logging onboard error-stats** command to display onboard error statistics. The output includes the following counters:

- FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER
- FCP_CNTR_MAC_RX_EOFA

- FCP_CNTR_MAC_RX_CRC
- FCP_CNTR_MAC_RX_MAX_FRAME_TRUNCATE
- FCP_CNTR_MAC_RX_MIN_FRAME_PAD
- FCP_CNTR_CREDIT_LOSS
- FCP_CNTR_TX_WT_AVG_B2B_ZERO

The following is an example outout of the **show logging onboard error-stats** command:

```
switch# show logging onboard error-stats
----------------------------
    Module:  1
----------------------------


------------------------------------------------------------------------------------------
ERROR STATISTICS INFORMATION FOR DEVICE: FCMAC
------------------------------------------------------------------------------------------
Interface Range    |                                        |         |Time Stamp
                   |            Error Stat Counter Name     |  Count  |MM/DD/YY HH:MM:SS
                   |                                        |         |
------------------------------------------------------------------------------------------
fc1/9              |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |4        |11/15/19 09:54:40
fc1/33             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |4        |11/15/19 09:37:53
fc1/36             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |4        |11/15/19 09:05:13
fc1/37             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |4        |11/15/19 08:42:56
fc1/37             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |4        |11/15/19 08:21:19
fc1/28             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |4        |11/15/19 08:20:59
fc1/9              |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |5996     |11/14/19 10:25:45
fc1/9              |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |5992     |11/14/19 06:19:04
fc1/36             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |22112    |11/14/19 06:19:04
fc1/36             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |21876    |11/14/19 06:18:44
fc1/36             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |21368    |11/14/19 06:18:24
fc1/36             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |20872    |11/14/19 06:18:04
fc1/36             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |20292    |11/14/19 06:17:44
fc1/36             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |19720    |11/14/19 06:17:24
fc1/36             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |19284    |11/14/19 06:17:04
fc1/36             |FCP_CNTR_MAC_RX_BAD_WORDS_FROM_DECODER  |18788    |11/14/19 06:16:44
```

Using Diagnostics

Generic online diagnostics (GOLD) define a common framework for diagnostic operations across Cisco platforms. The GOLD implementation checks the health of hardware components and verifies proper operation of the system data and control planes. Some tests take effect when the system is booting up; other tests take effect when the system is operational. A booting module goes through a series of checks before coming online to allow the system to detect faults in the hardware components at bootup and to ensure that a failing module is not introduced in a live network.

Defects are also diagnosed during system operation or runtime. You can configure a series of diagnostic checks to determine the condition of an online system. You must distinguish between disruptive and nondisruptive diagnostic tests. Although nondisruptive tests occur in the background and do not affect the system data or control planes, disruptive tests do affect live packet flows. You should schedule disruptive tests during special maintenance windows. The **show diagnostic content module** command output displays test attributes such as disruptive or nondisruptive tests.

You can configure runtime diagnostic checks to run at a specific time or to run continually in the background.

Health-monitoring diagnostic tests are nondisruptive, and they run in the background while the system is in operation. The role of online diagnostic health monitoring is to proactively detect hardware failures in the live network environment and inform you of a failure.

GOLD collects diagnostic results and detailed statistics for all tests including the last execution time, the first and last test pass time, the first and last test failure time, the total run count, the total failure count, the consecutive failure count, and the error code. These test results help administrators determine the condition of a system and understand the reason for a system failure. Use the **show diagnostic result** command to view diagnostic results.

For more information about configuring GOLD, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

Using Embedded Event Manager

Embedded Event Manager (EEM) is a policy-based framework that allows you to monitor key system events and then act on those events through a set policy. The policy is a preprogrammed script that you can load that defines actions that the device should invoke based on set events occurring. The script can generate actions, including, but not limited to, generating custom syslog or SNMP traps, invoking CLI commands, forcing a failover, and much more.

For more information about configuring EEM, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

Using Ethanalyzer

Ethanalyzer is a Cisco NX-OS protocol analyzer tool implementation of the open source software TShark which is a terminal version of Wireshark (formerly Ethereal). You can use Ethanalyzer to troubleshoot your network by capturing and analyzing control-plane traffic on inband and management interfaces across all Nexus platforms.

Note Performing ethanalyzer using the **front-panel** option for an interface bundled into a port-channel is not supported. Instead, use the **port-channel** option.

Beginning with Cisco NX-OS Release 10.3(1)F, Ethanalyzer is supported on Cisco Nexus 9808 platform switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, Ethanalyzer is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.

Beginning with Cisco NX-OS Release 10.4(1)F, Ethanalyzer is supported on Cisco Nexus 9804 platform switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.

To configure Ethanalyzer, use the following commands:

Command	Purpose
ethanalyzer local interface inband	Captures packets sent or received by the supervisor through the inband interface and displays summarized protocol information for captured packets.

Command	Purpose
ethanalyzer local interface inband-in	Captures packets received by the supervisor through the inband interface and displays summarized protocol information for captured packets.
ethanalyzer local interface inband-out	Captures packets sent by the supervisor through the inband interface and displays summarized protocol information for captured packets.
ethanalyzer local interface mgmt	Captures packets sent or received by the management interface and displays summarized protocol information for captured packets.
ethanalyzer local interface front-panel	Captures packets sent or received by the supervisor through a Layer 3 (routed) front-panel port and displays summarized protocol information for captured packets. **Note** This command does not support capturing packets sent or received by the supervisor through Layer 2 (switchport) front-panel ports.
ethanalyzer local interface port-channel	Captures packets sent or received by the supervisor through a Layer 3 (routed) port-channel interface and displays summarized protocol information for captured packets. **Note** This command does not support capturing packets sent or received by the supervisor through Layer 2 (switchport) port-channel interfaces.
ethanalyzer local interface vlan	Captures packets sent or received by the supervisor through a Layer 3 Switch Virtual Interface (SVI) and displays summarized protocol information.
ethanalyzer local interface netstack	Captures packets sent or received by the supervisor through the Netstack software component and displays summarized protocol information.
ethanalyzer local interface {**front-panel** | **inband** | **inband-in** | **inband-out** | **mgmt** | **port-channel** | **vlan**} **limit-captured-frames**	Limits the number of frames to capture within the Ethanalyzer session. The number of frames can be an integer value from 0 to 500,000. If 0 is provided, then a maximum of 500,000 frames will be captured before the Ethanalyzer session automatically stops.
ethanalyzer local interface {**front-panel** | **inband** | **inband-in** | **inband-out** | **mgmt** | **port-channel** | **vlan**} **limit-frame-size**	Limits the length of the frame to capture. The length of frame can be an integer value from 192 to 65,536.
ethanalyzer local interface {**front-panel** | **inband** | **inband-in** | **inband-out** | **mgmt** | **port-channel** | **vlan**} **capture-filter**	Filters the types of packets to capture using Berkeley Packet Filter (BPF) syntax.

Command	Purpose
ethanalyzer local interface {**front-panel** | **inband** | **inband-in** | **inband-out** | **mgmt** | **port-channel** | **vlan**} **display-filter**	Filtersthe types of captured packets to display using Wireshark or TShark Display Filters.
ethanalyzer local interface {**front-panel** | **inband** | **inband-in** | **inband-out** | **mgmt** | **port-channel** | **vlan**} **write**	Saves the captured data to a file. Valid storage options include the switch's bootflash, logflash, a USB storage device, or volatile storage.
ethanalyzer local read	Opens a captured data file and analyzes the file. Valid storage options include the switch's bootflash, logflash, a USB storage device, or volatile storage.
ethanalyzer local interface {**front-panel** | **inband** | **inband-in** | **inband-out** | **mgmt** | **port-channel** | **vlan**} **autostop**	Specifies a condition that will automatically stop the Ethanalyzer session. You can specify the duration of the session in seconds, number of files to capture when writing captured packets to a file using the **write** keyword, and file size when writing captured packets to a file using the **write** keyword.
ethanalyzer local interface {**front-panel** | **inband** | **inband-in** | **inband-out** | **mgmt** | **port-channel** | **vlan**} **capture-ring-buffer**	Specifies the capture ring buffer options for Ethanalyzer. This option will continuously write to one or more files in a ring buffer when combined with the write keyword. You can specify the duration in seconds that Ethanalyzer will wait before writing to a new file, the number of files to keep as part of the ring buffer, and the file size of each individual file in the ring buffer.
ethanalyzer local interface {**front-panel** | **inband** | **inband-in** | **inband-out** | **mgmt** | **port-channel** | **vlan**} **detail**	Displays detailed protocol information for captured packets.
ethanalyzer local interface {**front-panel** | **inband** | **inband-in** | **inband-out** | **mgmt** | **port-channel** | **vlan**} **raw**	Displays captured packets inhex format.
ethanalyzer local interface {**front-panel** | **inband** | **inband-in** | **inband-out** | **mgmt** | **port-channel** | **vlan**} **vrf**	Specifies the VRF that the Layer 3 interface is a member if the Layer 3 interface is in a non-default VRF.

Guidelines and Limitations

- If a Layer 3 interface is a member of a non-default VRF and is specified in an Ethanalyzer session (for example, through the **ethanalyzer local interface front-panel ethernet1/1** or **ethanalyzer local interface port-channel1** commands), you must specify the VRF that the Layer 3 interface is a member of within the Ethanalyzer session using the **vrf** keyword. For example, to capture packets received or sent by the supervisor through Layer 3 front-panel port Ethernet1/1 in VRF "red", use the **ethanalyzer local interface front-panel ethernet1/1 vrf red** command.
- When writing to a file, Ethanalyzer will automatically stop if the Ethanalyzer session captures 500,000 packets, or if the size of the file reaches ~11 megabytes, whichever comes first.

Examples

```
switch(config)# ethanalyzer local interface inband
<CR>
> Redirect it to a file
>> Redirect it to a file in append mode
autostop Capture autostop condition
capture-filter Filter on ethanalyzer capture capture-ring-buffer Capture ring buffer option
decode-internal Include internal system header decoding detail Display detailed protocol
information
display-filter Display filter on frames captured
limit-captured-frames Maximum number of frames to be captured (default is 10) limit-frame-size
 Capture only a subset of a frame
mirror Filter mirrored packets
raw Hex/Ascii dump the packet with possibly one line summary
write Filename to save capture to
| Pipe command output to filter
```

```
switch(config)# ethanalyzer local interface inband Capturing on 'ps-inb'

1 2021-07-26 09:36:36.395756813 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 64 PRI:
7 DEI: 0 ID: 4033
2 2021-07-26 09:36:36.395874466 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 205 PRI:
7 DEI: 0 ID: 4033
4 3 2021-07-26 09:36:36.395923840 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 806 PRI:
7 DEI: 0 ID: 4033
4 2021-07-26 09:36:36.395984384 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 1307 PRI:
7 DEI: 0 ID: 4033
5 2021-07-26 09:37:36.406020552 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 64 PRI:
7 DEI: 0 ID: 4033
6 2021-07-26 09:37:36.406155603 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 205 PRI:
7 DEI: 0 ID: 4033
7 2021-07-26 09:37:36.406220547 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 806 PRI:
7 DEI: 0 ID: 4033
8 8 2021-07-26 09:37:36.406297734 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 1307
PRI: 7 DEI: 0 ID: 4033
9 2021-07-26 09:38:36.408983263 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 64 PRI:
7 DEI: 0 ID: 4033
10 10 2021-07-26 09:38:36.409101470 00:22:bd:cf:b9:01 → 00:22:bd:cf:b9:00 0x3737 205
PRI: 7 DEI: 0 ID: 4033
```

Use the `detail` option for detailed protocol information. Ctrl+C can be used to abort and get the switch prompt back in the middle of the capture, if required.

```
switch(config)# ethanalyzer local interface inband detail
Capturing on 'ps-inb'
Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface ps-inb, id
 0
Interface id: 0 (ps-inb) Interface name: ps-inb
Encapsulation type: Ethernet (1)
Arrival Time: Jul 26, 2021 11:54:37.155791496 UTC
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1627300477.155791496 seconds
[Time delta from previous captured frame: 0.000000000 seconds] [Time delta from previous
displayed frame: 0.000000000 seconds] [Time since reference or first frame: 0.000000000
seconds] Frame Number: 1
Frame Length: 64 bytes (512 bits)
Capture Length: 64 bytes (512 bits) [Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:vlan:ethertype:data] Ethernet II, Src: 00:22:bd:cf:b9:01,
 Dst: 00:22:bd:cf:b9:00
Destination: 00:22:bd:cf:b9:00 Address: 00:22:bd:cf:b9:00
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
```

```
.... ...0 .... .... .... .... = IG bit: Individual address (unicast) Source: 00:22:bd:cf:b9:01
Address: 00:22:bd:cf:b9:01
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast) Type: 802.1Q Virtual
LAN (0x8100)
802.1Q Virtual LAN, PRI: 7, DEI: 0, ID: 4033
111. .... .... .... = Priority: Network Control (7) 4 ...0 .... .... .... = DEI: Ineligible
.... 1111 1100 0001 = ID: 4033
Type: Unknown (0x3737) Data (46 bytes)

0000 a9 04 00 00 7d a2 fe 60 47 4f 4c 44 00 0b 0b 0b ....}..`GOLD....
0010 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b ................


0020 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b ..............
Data: a90400007da2fe60474f4c44000b0b0b0b0b0b0b0b0b… [Length: 46]
```

Use the `capture-filter` option to select which packets to display or save to disk during capture. A capture filter maintains a high rate of capture while it filters. Because full dissection has not been done on the packets, the filter fields are predefined and limited.

Use the `display-filter` option to change the view of a capture file. A display filteruses fully dissected packets, so you can do very complex and advanced filtering when you analyze a network tracefile. Ethanalyzer writes captured data to a temporary file if it is not instructed to write captured data to a file elsewhere. This temporary file can fill quickly when a display filter is used without the user's knowledge, since all packets matching the `capture-filter` option are written to the temporary file, but only packets matching the `display-filter` option are displayed.

In this example, `limit-captured-frames` is set to 5. With the `capture-filter` option, Ethanalyzer shows you five packets which match the filter `host 10.10.10.2`. With the `display-filter` option, Ethanalyzer first captures five packets then displays only the packets that match the filter `ip.addr==10.10.10.2`.

```
switch(config)# ethanalyzer local interface inband capture-filter "host 10.10.10.2"
limit-captured-frames 5
Capturing on inband
2013-02-10 12:51:52.150404 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination port:
3200
2013-02-10 12:51:52.150480 10.10.10.2 -> 10.10.10.1 UDP Source port: 3200 Destination port:
3200
2013-02-10 12:51:52.496447 10.10.10.2 -> 10.10.10.1 UDP Source port: 3200 Destination port:
3200
2013-02-10 12:51:52.497201 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination port:
3200
2013-02-10 12:51:53.149831 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination port:
3200
5 packets captured
switch(config)# ethanalyzer local interface inband display-filter "ip.addr==10.10.10.2"
limit-captured-frame 5
Capturing on inband
2013-02-10 12:53:54.217462 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination port:
3200
2013-02-10 12:53:54.217819 10.10.10.2 -> 10.10.10.1 UDP Source port: 3200 Destination port:
3200
2 packets captured
```

The `write` option lets you write the capture data to a file in one of the storage devices (such as boothflash or logflash) on the Cisco Nexus 9000 Series Switch for later analysis. The capture file size is limited to 10 MB.

An example Ethanalyzer command with a `write` option is **ethanalyzer local interface inband writebootflash:***capture_file_name*. The following is an example of a `write` option with `capture-filter` and an output file name of `first-capture`:

```
switch(config)# ethanalyzer local interface inband capture-filter "host 10.10.10.2"
limit-captured-frame 5 write ?
bootflash: Filename logflash: Filename slot0:     Filename
usb1:      Filename
usb2: Filename volatile: Filename
switch(config)# ethanalyzer local interface inband capture-filter "host 10.10.10.2"
limit-captured-frame 5 write bootflash:first-capture
```

When the capture data is saved to a file, the captured packets are, by default, not displayed in the terminal window. The **display** option forces Cisco NX-OS to display the packets while it saves the capture data to a file.

The **capture-ring-buffer** option creates multiple files after a specified number of seconds, a specified number of files, or a specified file size. The following are the definitions of those options:

```
switch(config)# ethanalyzer local interface inband capture-ring-buffer ?
duration Stop writing to the file or switch to the next file after value seconds have elapsed
files Stop writing to capture files after value number of files were written or begin again
 with the first file after value number of files were
written (form a ring buffer)
filesize Stop writing to a capture file or switch to the next file after it reaches a size
 of value kilobytes
```

The **read** option lets you read the saved file on the device itself.

```
switch(config)# ethanalyzer local read bootflash:first-capture
2013-02-10 12:51:52.150404 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination port:
3200
2013-02-10 12:51:52.150480 10.10.10.2 -> 10.10.10.1 UDP Source port: 3200 Destination port:
3200
2013-02-10 12:51:52.496447 10.10.10.2 -> 10.10.10.1 UDP Source port: 3200 Destination port:
3200
2013-02-10 12:51:52.497201 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination port:
3200
2013-02-10 12:51:53.149831 10.10.10.1 -> 10.10.10.2 UDP Source port: 3200 Destination port:
3200

switch(config)# ethanalyzer local read bootflash:first-capture detail Frame 1 (110 bytes
on wire, 78 bytes captured)
-------------------------------SNIP-----------------------------------------------
[Frame is marked: False]
[Protocols in frame: eth:ip:udp:data]
Ethernet II Src: 00:24:98:6f:ba:c4 (00:24:98:6f:ba:c4), Dst: 00:26:51:ce:0f:44
(00:26:51:ce:0f:44)
Destination: 00:26:51:ce:0f:44 (00:26:51:ce:0f:44) Address: 00:26:51:ce:0f:44
(00:26:51:ce:0f:44)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default) Source:
00:24:98:ce:6f:ba:c4 (00:24:98:6f:ba:c4)
Address: 00:24:98:6f:ba:c4 (00:24:98:6f:ba:c4)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default) Type: IP
 (0x0800)
Internet Protocol, Src: 10.10.10.1 (10.10.10.1), Dst: 10.10.10.2 (10.10.10.2)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0xc0 (DSC) 0x30: Class Selector 6; ECN: 0x00)
-------------------------------SNIP-----------------------------------------------
```

You can also transfer the file to a server or a PC and read it with Wireshark or any other application that can read files with .cap or .pcap file formats.

```
switch(config)# copy bootflash:first-capture tftp:
Enter vrf (If no input, current vrf 'default' is considered): management
Enter hostname for the tftp server: 192.168.21.22
```

```
Trying to connect to tftp server......
Connection to Server Established. TFTP put operation was successful
Copy complete.
```

The **decode-internal** option reports internal information on how the Nexus 9000 forwards the packet. This information helps you understand and troubleshoot the flow of packets through the CPU.

```
switch(config)# ethanalyzer local interface inband decode-internal capture-filter "host
10.10.10.2" limit-captured-frame 5 detail
Capturing on inband NXOS Protocol
NXOS VLAN: 0====================->VLAN in decimal=0=L3 interface
NXOS SOURCE INDEX: 1024 ====================->PIXN LTL source index in decimal=400=SUP
inband
NXOS DEST INDEX: 2569====================-> PIXN LTL destination index in decimal=0xa09=e1/25
 Frame 1: (70 bytes on wire, 70 bytes captured)
Arrival Time: Feb 10, 2013 22:40:02.216492000
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1627300477.155791496 seconds
[Time delta from previous captured frame: 0.000000000 seconds] [Time delta from previous
displayed frame: 0.000000000 seconds] [Time since reference or first frame: 0.000000000
seconds] Frame Number: 1
Frame Length: 70 bytes Capture Length: 70 bytes [Frame is marked: False]
[Protocols in frame: eth:ip:udp:data]
Ethernet II, Src: 00:26:51:ce:0f:43 (00:26:51:ce:0f:43), Dst: 00:24:98:6f:ba:c3
(00:24:98:6f:ba:c3)
Destination: 00:24:98:6f:ba:c3 (00:24:98:6f:ba:c3) Address: 00:24:98:6f:ba:c3
(00:24:98:6f:ba:c3)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default) Source:
00:26:51:ce:0f:43 (00:26:51:ce:0f:43)
------------------------------SNIP---------------------------------------------
```

Convert the NX-OS index to hexadecimal, then use the **show system internal pixm info ltl {index}** command to map the local target logic (LTL) index to a physical or logical interface.

Capture Traffic to or from an IP Host

```
host 1.1.1.1
```

Capture Traffic to or from a Range of IP Addresses

```
net 172.16.7.0/24
```

net 172.16.7.0 mask 255.255.255.0

Capture Traffic from a Range of IP Addresses

```
src net 172.16.7.0/24
```

src net 172.16.7.0 mask 255.255.255.0

Capture Traffic to a Range of IP Addresses

```
dst net 172.16.7.0/24
```

dst net 172.16.7.0 mask 255.255.255.0

Capture UDLD, VTP, or CDP Traffic

UDLD is Unidirectional Link Detection, VTP is the VLAN Trunking Protocol, and CDP is the Cisco Discovery Protocol.

```
ether host 01:00:0c:cc:cc:cc
```

Capture Traffic to or from a MAC Address

```
ether host 00:01:02:03:04:05
```

Note

and = &&

or = ||

not = !

MAC address format : xx:xx:xx:xx:xx:xx

Common Control Plane Protocols

- UDLD: Destination Media Access Controller (DMAC) = 01-00-0C-CC-CC-CC and EthType = 0x0111
- LACP: DMAC = 01:80:C2:00:00:02 and EthType = 0x8809. LACP stands for Link Aggregation Control Protocol
- STP: DMAC = 01:80:C2:00:00:00 and EthType = 0x4242 - or - DMAC = 01:00:0C:CC:CC:CD and EthType = 0x010B
- CDP: DMAC = 01-00-0C-CC-CC-CC and EthType = 0x2000
- LLDP: DMAC = 01:80:C2:00:00:0E or 01:80:C2:00:00:03 or 01:80:C2:00:00:00 and EthType = 0x88CC
- DOT1X: DMAC = 01:80:C2:00:00:03 and EthType = 0x888E. DOT1X stands for IEEE 802.1x
- IPv6: EthType = 0x86DD
- List of UDP and TCP port numbers

Ethanalyzer does not capture data traffic that Cisco NX-OS forwards in the hardware.

Ethanalyzer uses the same capture filter syntax as **tcpdump** and uses the Wireshark display filter syntax.

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

```
switch(config)# ethanalyzer local interface mgmt limit-captured-frames 4
Capturing on eth1

2013-05-18 13:21:21.841182 172.28.230.2 -> 224.0.0.2 BGP Hello (state Standy)
2013-05-18 13:21:21.842190 10.86.249.17 -> 172.28.231.193 TCP 4261 > telnet [AC] Seq=0 Ack=0
 Win=64475 Len=0
2013-05-18 13:21:21.843039 172.28.231.193 -> 10.86.249.17 TELNET Telnet Data ..
2013-05-18 13:21:21.850463 00:13:5f:1c:ee:80 -> ab:00:00:02:00:00 0x6002 DEC DN

Remote Console
4 packets captured
```

This example shows detailed captured data for one HSRP packet:

```
switch(config)# ethanalyzer local interface mgmt capture-filter "udp port 1985"
```

```
limit-captured-frames 1
Capturing on eth1
Frame 1 (62 bytes on wire, 62 bytes captured)
Arrival Time: May 18, 2013 13:29:19.961280000
[Time delta from previous captured frame: 1203341359.961280000 seconds]
[Time delta from previous displayed frame: 1203341359.961280000 seconds]
[Time since reference or first frame: 1203341359.961280000 seconds]
Frame Number: 1
Frame Length: 62 bytes
Capture Length: 62 bytes
[Frame is marked: False]
[Protocols in frame: eth:ip:udp:hsrp]

Ethernet II, Src: 00:00:0c:07:ac:01 (00:00:0c:07:ac:01), Dst: 01:00:5e:00:00:02
(01:00:5e:00:00:02)
Destination: 01:00:5e:00:00:02 (01:00:5e:00:00:02)
Address: 01:00:5e:00:00:02 (01:00:5e:00:00:02)
.... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
Source: 00:00:0c:07:ac:01 (00:00:0c:07:ac:01)
Address: 00:00:0c:07:ac:01 (00:00:0c:07:ac:01)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

Type: IP (0x0800)
Internet Protocol, Src: 172.28.230.3 (172.28.230.3), Dst: 224.0.0.2 (224.0.0.2)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00)
1100 00.. = Differentiated Services Codepoint: Class Selector 6 (0x30)
.... ..0. = ECN-Capable Transport (ECT): 0
.... ...0 = ECN-CE: 0

Total Length: 48
Identification: 0x0000 (0)
Flags: 0x00
0... = Reserved bit: Not set
.0.. = Don't fragment: Not set
..0. = More fragments: Not set
Fragment offset: 0
Time to live: 1
Protocol: UDP (0x11)
Header checksum: 0x46db [correct]
[Good: True]
[Bad : False]

Source: 172.28.230.3 (172.28.230.3)
Destination: 224.0.0.2 (224.0.0.2)
User Datagram Protocol, Src Port: 1985 (1985), Dst Port: 1985 (1985)
Source port: 1985 (1985)
Destination port: 1985 (1985)
Length: 28
Checksum: 0x8ab9 [correct]
[Good Checksum: True]
[Bad Checksum: False]

Cisco Hot Standby Router Protocol
Version: 0
Op Code: Hello (0)
State: Active (16)
Hellotime: Default (3)
Holdtime: Default (10)
Priority: 105
```

```
Group: 1
Reserved: 0Authentication Data: Default (cisco)
Virtual IP Address: 172.28.230.1 (172.28.230.1)

1 packets captured
```

This example uses a display filter to show only those HSRP packets that have an active HSRP state:

```
switch(config)# ethanalyzer local interface mgmt display-filter "hsrp.state==Active"
limit-captured-frames 2
Capturing on eth1

2013-05-18 14:35:41.443118 172.28.230.3 -> 224.0.0.2 HSRP Hello (state Active)
2013-05-18 14:35:44.326892 172.28.230.3 -> 224.0.0.2 HSRP Hello (state Active)
2 packets captured
```

Ethanalyzer Background Capture Process and Autocollection of Inband Packets

Ethanalyzer can be enabled to run as a background task to capture inband packets. The inband packet data is kept in RAM memory in PCAP files. A configurable limited amount of PCAP data (configurable number of files with a configurable file size) is available at any time. When the limit is reached, the oldest file gets overwritten with the current capture in a cyclic way.

The data captured by the Ethanalyzer background task is in RAM, and cyclically overwritten, not taking up bootflash space. For the user to be able to look at the data, a snapshot needs to be taken, which will copy the packet capture info taken by the background process in PCAP format from RAM to non-volatile storage (bootflash) for viewing. Users need to consider available bootflash space when taking a snapshot.

The snapshot can be triggered manually by the user via CLI. EEM policies, can be as well used to trigger the snapshot upon certain events. Use case examples of triggers are inband rate exceeding a defined threshold, CoPP drops exceeding a threshold - the snapshot gives insight what packets were hitting the inband upto the moment of the event.

When monitoring rates, a threshold that exceeds the normally expected or accepted rates by the user, needs to be set to avoid an excess of alerts for non-issues.Attention need to be paid when increasing the max-triggers in the autocollection EEM policy below. Not following these practices can result an excessive amount of irrelevant PCAP data to be snapshotted with a potential of filling up the bootflash.

Ethanalyzer has added CLIs to enable and configure the background session, start and stop the session, snapshot the Ethanalyzer information as well as show commands to look at the background session status. All CLIs are to be run from enable mode.

Guidelines and Limitations for Ethanalyzer Background Capture Process

- The Ethanalyzer background process stores .pcap files in the /tmp directory, which has limited storage capacity. Users must ensure that the total size of all `.pcap` files does not exceed the available `/tmp` storage space.

  To calculate the total space required for Ethanalyzer `.pcap` files, use the following formula:

  fileSize * numFiles < Available /tmp Space

  Before starting the Ethanalyzer background process, validate /tmp storage availability using the following command:

```
bash-4.4# df -k /tmp
Filesystem     1K-blocks  Used Available Use% Mounted on
none              614400  2760    611640   1% /var/volatile/tmp
```

- Restarting the Ethanalyzer background session deletes all previously captured .pcap files in /tmp. Users must copy important data to persistent storage (e.g., /bootflash) using the ethanalyzer copy-background-snapshot command before restarting.

  Use the following command to copy .pcap files to bootflash before restarting:

  ```
  ethanalyzer copy-background-snapshot
  ```

- Consider available bootflash space before taking a snapshot, as snapshots are copied from /tmp to bootflash. Insufficient bootflash space may result in failed snapshots or incomplete data storage.

  To save space, use the compressed tar option:

  ```
  ethanalyzer copy-compressed-background-snapshot
  ```

- Use Event Manager (EEM) policies to trigger snapshots based on events like inband rate thresholds or CoPP drops. Configure the max-triggers parameter carefully to avoid excessive alerts or irrelevant data snapshots, which could fill up bootflash unnecessarily.
- The background process does not automatically manage storage rollover beyond /tmp. Users must configure parameters properly to avoid filling up /tmp storage.
- If /tmp storage becomes full, collected data may be lost.

Table 6: Ethanalyzer CLIs

CLI	Description
ethanalyzer background-session config <filename|filesize|numfiles|session>	Configure parameters of ethanalyzer background process/session for capturinig packet in circular buffer. • Filename - Background packet capture file name saved by Ethanalyzer background capture process. • Filesize - Size of individual capture file that are in temporary buffer. Value ranges from 1-65536 KB. • Numfiles - Number of maximum pcap files to be stored in temporary buffer. Value ranges from 2-16. • Session – Enable/Disable Ethanalyzer background capture session.
ethanalyzer background-session restart	Start/Restart Ethanalyzer background capture session.
ethanalyzer background-session stop	Stops the Ethanalyzer background capture session.
show ethanalyzer background-session processes	Show Ethanalyzer background capture session details.
show ethanalyzer background-session config	Print Ethanalyzer background capture session configuration file.

CLI	Description
ethanalyzer copy-background-snapshot	Copy the files captured in temporary buffer to bootflash. Files are in pcap format.
ethanalyzer copy-compressed-background-snapshot	Tar the files captured in temporary buffer and copy the tar file to bootflash. **Note** Issuing this CLI multiple times will delete the old tar file. User discretion is adviced to copy the old tar file if it exists in bootflash.

Beginning with Cisco NX-OS, release 10.1(2) Ethanalyzer Autocollection CLI is supported on all Cisco Nexus 9000 Series platforms.

Ethanalyzer Autocollection CLI Warnings

The following are the Ethnalyzer Autocollection CLI warnings:

- Whenever any change is made to the background process, user is required ot restart/start the Ethanalyzer background process. The following warning message shall be displayed to user when any config change is made:

  “Please restart the Ethanalyzer background process for any config change to take effect.”

- In the platforms where supervisor redundancy is supported, switchover of the active supervisor can lead to the Ethanalyzer background capture process to fail to start automatically. User must manually restart the Ethanalyzer background process. If the user wants the Ethanalyzer background process to start automatically after switchover, the user must configure the session enable on the active supervisor, and then reload the switch to take effect. After this, even if the switchover occurs, the Ethanalyzer background capture process will start automatically in the newly active supervisor.

CLI Examples

Example CLI Output - All commands are run from enable mode

Step 1: Enable Ethanalyzer session running in the background:

```
switch# ethanalyzer background-session config session enable
```

```
switch# dir bootflash: | include dump
       1087    Jan 29 13:55:46 2021  dumpcap_bg_session_configuration.xml
switch# show ethanalyzer background-session config
<?xml version="1.0"?>
<!-- This document contains configuration settings for background packet -->
<!-- capture session to execute in ring buffer mode. Please modify the settings
based on system resources -->
<!-- path:         background packet capture directory where ring buffer files w
ill be saved -->
<!-- filename:     background packet capture file name saved by dumpcap. Files w
ill be generated as filename_number_date format -->
<!-- filesize:     Size of individual ring buffer file in kB. Note that the file
size is limited to a maximum value of 65536 kB-->
<!-- num_of_files: value begin again with the first file after value number of f
iles were written (form a ring buffer). The maximum value should be equal to 16
-->
<!-- session:      Enable/disable background packet capture session process. App
```

```
licable for both boot-up as well as session restart -->
<ethanalyzer_config>
    <filepath>/tmp/dumpcap_bg_session_files/</filepath>
    <filename>capture</filename>
    <filesize>2048</filesize>
    <numfiles>2</numfiles>
    <session>enable</session>
</ethanalyzer_config>
```

The following is the CLI output:

```
switch# ethanalyzer background-session restart
root     30038     1  0 13:58 ttyS0    00:00:00 /usr/bin/dumpcap -n -b filesize:
2048 -b files:2 -i ps-inb -Z none -w /tmp/dumpcap_bg_session_files/capture.pcap
```

Step 2: Verifying the background session configuration parameters

```
switch# show ethanalyzer background-session process
```

Step 3: Start the background Ethanalyzer process

```
switch# ethanalyzer background-session restart
```

Step 4: Verifying the running of Ethanalyzer background capture session

```
switch# ethanalyzer background-session processes
Background session of packet analyzer:
root 17216 1 4 12:43 ttyS0 00:00:00 /usr/bin/dumpcap -n -b filesize:2048 -b files:2 -i
ps-inb -Z none -w /tmp/dumpcap_bg_session_files/capture.pcap
switch#
```

Usecase example: Execute CLI to capture a snapshot for viewing

```
switch# ethanalyzer copy-background-snapshot

Copy packet analyzer captured frames to bootflash...
Copied snapshot files :
     72 -rw-rw-rw-  1 root  root               65844 Jan 21 00:21
CAPTURE_00001_20210121001903.pcap

switch# ethanalyzer copy-compressed-background-snapshot

Copy packet analyzer captured compressed frames to bootflash...
Copied snapshot files :
     28 -rw-r--r--  1 root  root               27181 Jan 21 00:22 CAPTURE.tar.gz
```

Usecase example: Using inband rate monitoring as a trigger for autocollection of Ethanalyzer snapshot.

Table 7: Inband Rate Monitoring CLI Options

CLI	Description
Config mode	system inband cpu-mac log threshold rx rx_pps tx tx_pps throttle secondsrx_pps, tx_pps: 0-1500000 Inband rx/tx pps rate that needs to be logged when exceededseconds: log throttle interval (maximum 1 exceed log per defined interval)
Enable mode	show system inband cpu-mac log threshold" to display settings
Default	off (PPS values 0), throttle interval 120 seconds.

The assumption is that the Ethanalyzer background process feature is configured and running as explained in the previous section. This usecase has example rates for demo or example purpose, but the user needs to use a realistic rate that is considered as worthwhile logging. A threshold that exceeds the user requirements needs to be notified to avoid an excess of alerts for non-issues.

Note Attention needs to be paid when increasing the max-triggers in the autocollection EEM policy below. Not following these practices can result an excessive amount of PCAP data to be snapshotted with a potential of filling up the bootflash.

The max-triggers parameter gets checked against the amount of snapshot files persistently stored in the eem_snapshots directory on bootflash (bootflash:eem_snapshots) of the active supervisor. In case of a supervisor switchover, the number of collections on the newly active supervisor can be different from what is on the previously active supervisor, resulting in autocollection to resume or not. The resuming of autocollection depends on the snapshot bundles present on the newly active supervisor's bootflash.

Once the amount of files in the directory mentioned matches max-triggers, autocollection will stop. To start it again, user must remove the snapshot files from the directory to bring the file count to a "value" lower than max-triggers, allowing for another amount (max-triggers minus "value") of autcollections. The details explained in the Trigger-Based Event Log Auto-Collection section of the Configuring the Embedded Event Manager chapter.

Step 1: Enable inband rate monitoring

```
switch(config)# system inband cpu-mac log threshold rx 400 tx 4000 throttle 60
switch# show system inband cpu-mac log threshold
Thresholds Rx: 400 PPS, Tx; 4000 PPS
Log throttle interval: 60 seconds
```

Leveraging the trigger based log file auto-collection, as explained in the Trigger-Based Event Log Auto-Collection section of the Configuring the Embedded Event Manager chapter, creating the directory (in the example below the name of the directory is "auto_collect") and creating or enabling the EEM policy, will enable the built-in snapshot collection of event logs and ethanalyzer pcap.

Step 2: Create the directory

```
create auto_collect directory
switch# pwd
bootflash:
switch# cd scripts
switch# mkdir auto_collect
```

Step 3: Enable the event manager policy

```
switch(config)# event manager applet syslog_trigger override __syslog_trigger_default

switch(config-applet)# action 1.0 collect auto_collect rate-limit 60 max-triggers 3
$_syslog_msg
```

This will enable autocollection for a max of 1x per 60 seconds, with a total max of 3 times for the same trigger, meaning we will store up to max-triggers x num_files pcap files for the same syslog trigger (in the example: 3 x 2 = 6 files).

The above use case in action: identifying a misbehaving host 20.1.1.100 launching high volume of ICMP request.

```
switch#
2021 Jan 29 15:15:27 switch %KERN-1-SYSTEM_MSG: [17181.984601] Inband Rx threshold 400 PPS
 reached. - kernel
```

```
2021 Jan 29 15:15:28 switch %KERN-1-SYSTEM_MSG: [17182.997911] Inband Rx threshold 400 PPS
 reached. - kernel
switch# show system internal event-logs auto-collect history
DateTime              Snapshot ID  Syslog
Status/Secs/Logsize(Bytes)
2021-Jan-29 15:15:30  620969861    KERN-1-SYSTEM_MSG
PROCESSED:1:7118865
2021-Jan-29 15:15:30  201962781    KERN-1-SYSTEM_MSG
DROPPED-LASTACTIONINPROG
2021-Jan-29 15:15:29  620969861    KERN-1-SYSTEM_MSG                        PROCESSING
...
switch# dir bootflash: | include capture
    2048040    Jan 29 15:15:29 2021  capture_00004_20210129150732.pcap
     169288    Jan 29 15:15:29 2021  capture_00005_20210129151528.pcap
 ...
```

To decode the file captured via background process, please contact cisco tac team.

Use case Example: Use a custom (non built in auto collection yaml) trigger (CoPP drop threshold exceed)

The following are the assumptions:

1. Ethanalyzer background process feature is configured and running as explained before.

2. Step 2 and Step 3 of the previous use case example are in place.

Enable CoPP threshold logging for the class interested in learning why drops happen. The details are in the CoPP configuration guide (reference).

In the example, for class copp-class-normal, which includes ARP, a threshold is set to 1000000 and the logging level is set to 1 (high enough to be picked up for autocollect):

```
class copp-class-normal
    logging drop threshold 1000000 level 1
```

In the same directory used in the previous use case example (bootflash:scripts/auto_collect), add file copp.yaml with the following (copp = the component name):

```
#**********************************************************************
#
# File:   comp specific yaml
# Author:
#
# Description: Module Makefile
#
#
# Copyright (c) 2019 by cisco Systems, Inc.
# All rights reserved.
#
#
# $Id: comp specific yaml $
# $Source:  $
# $Author:  $
#
#**********************************************************************
version: 1
components:
    copp:
            default:
            copp_drops1:
              serviceCOPP:
                match: CoPP drops exceed threshold
                commands: ethanalyzer copy-background-snapshot
```

The above use case in action: identifying high volume of ARP request causing CoPP drops in the class.

```
switch#
2021 Jan 29 15:49:47 switch %COPP-1-COPP_DROPS1: CoPP drops exceed threshold in class:
copp-class-normal-log,
check show policy-map interface control-plane for more info.
switch# show policy-map interface control-plane class copp-class-normal-log
Control Plane

  Service-policy  input: copp-policy-strict-log

    class-map copp-class-normal-log (match-any)
      match access-group name copp-acl-mac-dot1x-log
      match protocol arp
      set cos 1
      threshold: 1000000, level: 1
      police cir 1400 kbps , bc 32000 bytes
      module 1 :
        transmitted 25690204 bytes;
        5-minute offered rate 168761 bytes/sec
        conformed 194394 peak-rate bytes/sec
          at Fri Jan 29 15:49:56 2021

        dropped 92058020 bytes;
        5-min violate rate 615169 byte/sec
        violated 698977 peak-rate byte/sec        at Fri Jan 29 15:49:56 2021

switch#
switch# show system internal event-logs auto-collect history
DateTime              Snapshot ID  Syslog
Status/Secs/Logsize(Bytes)
2021-Jan-29 15:49:57  1232244872   COPP-1-COPP_DROPS1                       RATELIMITED
2021-Jan-29 15:49:50  522271686    COPP-1-COPP_DROPS1
PROCESSED:1:11182862
2021-Jan-29 15:49:48  522271686    COPP-1-COPP_DROPS1                       PROCESSING
...
switch# dir bootflash: | include capture
    2048192    Jan 29 15:49:49 2021  capture_00038_20210129154942.pcap
    1788016    Jan 29 15:49:49 2021  capture_00039_20210129154946.pcap
....
```

SSO Behavior

If standby supervisor comes up with background process config session=disable, then the user is expected to restart the process when this supervisor becomes active.

References

- Wireshark: CaptureFilters
- Wireshark: DisplayFilters
- Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide
- Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide
- Cisco Nexus 9000 NX-OS Interface Configuration Guide
- Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide

SNMP and RMON Support

Cisco NX-OS provides extensive SNMPv1, v2, and v3 support, including Management Information Bases (MIBs) and notifications (traps and informs).

The SNMP standard allows any third-party applications that support the different MIBs to manage and monitor Cisco NX-OS.

SNMPv3 provides extended security. Each device can be selectively enabled or disabled for SNMP service. In addition, each device can be configured with a method of handling SNMPv1 and v2 requests.

Cisco NX-OS also supports Remote Monitoring (RMON) alarms and events. RMON alarms and events provide a mechanism for setting thresholds and sending notifications based on changes in network behavior.

The *Alarm Group* allows you to set alarms. Alarms can be set on one or multiple parameters within a device. For example, you can set an RMON alarm for a specific level of CPU utilization on a device. The *EventGroup* allows you to configure events that are actions to be taken based on an alarm condition. The types of events that are supported include logging, SNMP traps, and log-and-trap.

For more information about configuring SNMP and RMON, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

Using the PCAP SNMP Parser

The PCAP SNMP parser is a tool to analyze SNMP packets captured in .pcap format. It runs on the switch and generates a statistics report for all of the SNMP get, getnext, getbulk, set, trap, and response requests sent to the switch.

To use the PCAP SNMP parser, use one of the following commands:

- **debug packet-analysis snmp** [**mgmt0** | **inband**] **duration** *seconds* [*output-file*] [**keep-pcap**]—Captures packets for a specified number of seconds using Tshark, saves them in a temporary .pcap file, and then analyzes them based on this .pcap file.

  The results are saved in the output file or printed to the console, if the output file is not specified. The temporary .pcap file will be deleted by default, unless you use the **keep-pcap** option. Packet capture can be performed on the management interface (mgmt0), which is the default, or the inband interface.

  Examples:

  ```
  switch# debug packet-analysis snmp duration 100

  switch# debug packet-analysis snmp duration 100 bootflash:snmp_stats.log

  switch# debug packet-analysis snmp duration 100 bootflash:snmp_stats.log keep-pcap

  switch# debug packet-analysis snmp inband duration 100

  switch# debug packet-analysis snmp inband duration 100 bootflash:snmp_stats.log

  switch# debug packet-analysis snmp inband duration 100 bootflash:snmp_stats.log keep-pcap
  ```

- **debug packet-analysis snmp** *input-pcap-file* [*output-file*]—Analyzes the captured packets on an existing .pcap file.

Examples:

```
switch# debug packet-analysis snmp bootflash:snmp.pcap

switch# debug packet-analysis snmp bootflash:snmp.pcap bootflash:snmp_stats.log
```

The following example shows a sample statistics report for the **debug packet-analysis snmp** [**mgmt0** | **inband**] **duration** command:

```
switch# debug packet-analysis snmp duration 10
Capturing on eth0
36
wireshark-cisco-mtc-dissector: ethertype=0xde09, devicetype=0x0
wireshark-broadcom-rcpu-dissector: ethertype=0xde08, devicetype=0x0

Started analyzing. It may take several minutes, please wait!

Statistics Report
-----------------------------------------
SNMP Packet Capture Duration: 0 seconds
Total Hosts: 1
Total Requests: 18
Total Responses: 18
Total GET: 0
Total GETNEXT: 0
Total WALK: 1 (NEXT: 18)
Total GETBULK: 0
Total BULKWALK: 0 (BULK: 0)
Total SET: 0
Total TRAP: 0
Total INFORM: 0

Hosts         GET  GETNEXT  WALK(NEXT) GETBULK  BULKWALK(BULK) SET  TRAP  INFORM  RESPONSE
------------------------------------------------------------------------------------------
10.22.27.244   0     0        1(18)      0         0(0)         0    0      0       18

Sessions
--------
1

MIB Objects GET  GETNEXT  WALK(NEXT) GETBULK(Non_rep/Max_rep) BULKWALK(BULK, Non_rep/Max_rep)
------------------------------------------------------------------------------------------
ifName      0       0        1(18)      0                        0


SET     Hosts
--------------------
0       10.22.27.244
```

Using RADIUS

The RADIUS protocol is used to exchange 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 NX-OS device. When you try to log into a device, Cisco NX-OS 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 device can then refer to the RADIUS server to determine the access that the user will have.

Accounting refers to the log information that is kept for each management session in a device. You can use this information to generate reports for troubleshooting purposes and user accountability. You can implement accounting locally or remotely (using RADIUS).

This example shows how to display accounting log entries:

```
switch# show accounting log
Sun May 12 04:02:27 2007:start:/dev/pts/0_1039924947:admin
Sun May 12 04:02:28 2007:stop:/dev/pts/0_1039924947:admin:vsh exited normally
Sun May 12 04:02:33 2007:start:/dev/pts/0_1039924953:admin
Sun May 12 04:02:34 2007:stop:/dev/pts/0_1039924953:admin:vsh exited normally
Sun May 12 05:02:08 2007:start:snmp_1039928528_172.22.95.167:public
Sun May 12 05:02:08 2007:update:snmp_1039928528_172.22.95.167:public:Switchname
```

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

Using 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
- Selection of the types of logging information to be captured
- Selection of the destination of the captured logging information

You can use syslog to store a chronological log of system messages locally or to send this information to a central syslog server. The 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.

The 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 device. For example, you might want to report debug events only for the OSPF service but record all severity level events for the BGP service.

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) are saved in NVRAM. You can view this log at any time with the **show logging nvram** command.

Logging Levels

Cisco NX-OS supports the following logging levels:

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

By default, the device logs normal but significant system messages to a log file and sends these messages to the system console. Users can specify which system messages should be saved based on the type of facility and the severity level. Messages have a time stamp 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.

- To disable console logging, use the **no logging console** command in configuration mode.
- To enable logging for Telnet or SSH, use the **terminal monitor** command in EXEC mode.
- When logging to a console session is disabled or enabled, that state is applied to all future console sessions. If a user exits and logs in again to a new session, the state is preserved. However, when logging to a Telnet or SSH session is enabled or disabled, that state is applied only to that session. The state is not preserved after the user exits the session.

The **no logging console** command disables console logging and is enabled by default.

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

The **terminal monitor** command enables logging for Telnet or SSH and is disabled by default.

```
switch# terminal monitor
```

For more information about configuring syslog, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

Using SPAN

You can use the Switched Port Analyzer (SPAN) utility to perform detailed troubleshooting or to take a sample of traffic from a particular application host for proactive monitoring and analysis.

When you have a problem in your network that you cannot solve by fixing the device configuration, you typically need to take a look at the protocol level. You can use **debug** commands to look at the control traffic between an end node and a device. However, when you need to focus on all the traffic that originates from or is destined to a particular end node, you can use a protocol analyzer to capture protocol traces.

To use a protocol analyzer, you must insert the analyzer inline with the device under analysis, which disrupts input and output (I/O) to and from the device.

In Ethernet networks, you can solve this problem by using the SPAN utility. SPAN allows you to take a copy of all traffic and direct it to another port within the device. The process is nondisruptive to any connected devices and is facilitated in the hardware, which prevents any unnecessary CPU load.

SPAN allows you to create independent SPAN sessions within the device. You can apply a filter to capture only the traffic received or the traffic transmitted.

To start the SPAN utility, use the **span session** *span-num* command where *span-num* identifies a specific SPAN session. When you enter this command, the system displays a submenu, which allows you to configure the destination interface and the source VLAN or interfaces.

```
switch2# config terminal
switch2(config)# span session 1 <<=== Create a span session
switch2(config-span)# source interface e1/8 <<=== Specify the port to be spanned
switch2(config-span)# destination interface e1/3 <<==== Specify the span destination port
switch2(config-span)# end
switch2# show span session 1
Session 1 (active)
Destination is e1/3
No session filters configured
Ingress (rx) sources are
e1/8,
Egress (tx) sources are
fe1/8,
```

For more information about configuring SPAN, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

SPAN Consistency Checker

SPAN Consistency Checker performs a check on the program and consistency configurations for Supervisor, Line cards, and Hardware tables. While configuring a SPAN on a switch, its state gets programed in software, storage, line card, and hardware tables. If these states are not in sync with each other, the SPAN session fails. The SPAN Consistency Checker helps in identifying the inconsistencies in a SPAN session that can be fixed instantly.

The **cc_monitor_session.py** is the python script for the SPAN Consistency Checker. This python script fetches the states on the Supervisor, Line cards, and Hardware tables and checks if all the states are in sync with each other.

The following is the CLI for SPAN Consistency Checker:

```
show consistency-checker monitor session {<session-id> | all}
```

This CLI executes the python script in the backend and displays the output of the SPAN Consistency Checker. The following is the output:

```
switch# show consistency-checker monitor session 1
Monitor Consistency Check : PASSED
```

Using sFlow

Sampled flow (sFlow) allows you to monitor real-time traffic in data networks that contain switches and routers. It uses the sampling mechanism in the sFlow agent software on switches and routers to monitor traffic and to forward the sample data to the central data collector. For more information about sFlow, see RFC 3176.

The sFlow agent, which is embedded in the Cisco NX-OS software, periodically samples or polls the interface counters that are associated with a data source of the sampled packets.

For more information about configuring sFlow, see Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

sFlow Consistency Checker

sFlow Consistency Checker performs a check on the program and consistency configurations for supervisor and line card hardware tables. While configuring sFlow on a switch, its state gets programmed in software, storage, and line card hardware tables. However, on Cisco Nexus 9808 switches, Consistency Checker performs a check on the program and consistency configurations for supervisor and line card hardware abstractions layer. While configuring sFlow on a switch, if the states are not in sync with each other, the SPAN session fails. The sFlow Consistency Checker helps in identifying the inconsistencies in an sFlow session that can be fixed instantly.

You can use the sFlow Consistency Checker to validate the consistency of configurations on the sFlow supervisor process.

Note The sFlow Consistency Checker validates the sFlow configuration information related to data source in sFlow process only.

The following is the command for sFlow Consistency Checker:

```
switch(config)# show consistency-checker sflow
```

The following is a sample output:

```
switch(config)# show consistency-checker sflow
SFLOW CC validation start:
passed for interface ethernet 1/15
Consistency checker passed for SFLOW
```

Using the Blue Beacon Feature

On some platforms, you can cause the platform LEDs to blink. This feature is a useful way to mark a piece of hardware so that a local administrator can quickly identify the hardware for troubleshooting or replacement.

To flash the LEDs on a hardware entity, use the following commands:

Command	Purpose
blink chassis	Flashes the chassis LED.

Command	Purpose
blink fan *number*	Flashes one of the fan LEDs.
blink module *slot*	Flashes the selected module LED.
blink powersupply *number*	Flashes one of the power supply LEDs.

Using the watch Command

The **watch** command allows you to refresh and monitor Cisco NX-OS CLI command output or Unix command output (through the **run bash** *command* command).

Use the command as follows:

watch [**differences**] [**interval** *seconds*] *command*watch

- **differences**—Highlights the differences in the command output.
- **interval** *seconds*—Specifies how often the command output is refreshed. The range is from 0 to 2147483647 seconds.
- *command*—Specifies the command that you want to watch.

The following example shows how the **watch** command can be used to refresh the output of the **show interface eth1/15 counters** command every second and to highlight any differences:

```
switch# watch differences interval 1 show interface eth1/15 counters

Every 1.0s:  vsh -c "show interface eth1/15 counters"      Mon Aug 31 15:52:53 2015


--------------------------------------------------------------------------------
Port                                 InOctets                      InUcastPkts
--------------------------------------------------------------------------------
Eth1/15                                583736                                0

--------------------------------------------------------------------------------
Port                              InMcastPkts                      InBcastPkts
--------------------------------------------------------------------------------
Eth1/15                                  2433                                0

--------------------------------------------------------------------------------
Port                                OutOctets                     OutUcastPkts
--------------------------------------------------------------------------------
Eth1/15                               5247672                                0

--------------------------------------------------------------------------------
Port                             OutMcastPkts                     OutBcastPkts
--------------------------------------------------------------------------------
Eth1/15                                 75307                                0
```

Additional References for Troubleshooting Tools and Methodology

Related Documents

Related Topic	Document Title
System management tools	*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*
MIBs	*Cisco Nexus 7000 Series and 9000 Series NX-OS MIB Quick Reference*

INDEX