Network Management Configuration Guide, Cisco IOS XE Release 3SE (Catalyst 3650 Switch)

First Published: October 10, 2013

Last Modified: April 15, 2016

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

Text Part Number: OL-30715-01

CONTENTS

Preface

Document Conventions

This document uses the following conventions:

Convention	Description
^ or Ctrl	Both the ^ symbol and Ctrl represent the Control (Ctrl) key on a keyboard. For example, the key combination **^D** or **Ctrl-D** means that you hold down the Control key while you press the D key. (Keys are indicated in capital letters but are not case sensitive.)
bold font	Commands and keywords and user-entered text appear in **bold** font.
Italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
`Courier` font	Terminal sessions and information the system displays appear in `courier` font.
`Bold Courier` font	**`Bold Courier`** font indicates text that the user must enter.
[x]	Elements in square brackets are optional.
...	An ellipsis (three consecutive nonbolded periods without spaces) after a syntax element indicates that the element can be repeated.
|	A vertical line, called a pipe, indicates a choice within a set of keywords or arguments.
[x | y]	Optional alternative keywords are grouped in brackets and separated by vertical bars.

Convention	Description
{x | y}	Required alternative keywords are grouped in braces and separated by vertical bars.
[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.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
< >	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.

Reader Alert Conventions

This document may use the following conventions for reader alerts:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Tip Means *the following information will help you solve a problem.*

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Timesaver Means *the described action saves time.* You can save time by performing the action described in the paragraph.

Warning **Means *reader be warned.* In this situation, you might perform an action that could result in bodily injury.**

Related Documentation

Note Before installing or upgrading the switch, refer to the switch release notes.

- Cisco Catalyst 3650 Switch documentation, located at:

  http://www.cisco.com/go/cat3650_docs

- Cisco SFP and SFP+ modules documentation, including compatibility matrixes, located at:

  http://www.cisco.com/en/US/products/hw/modules/ps5455/tsd_products_support_series_home.html

- Error Message Decoder, located at:

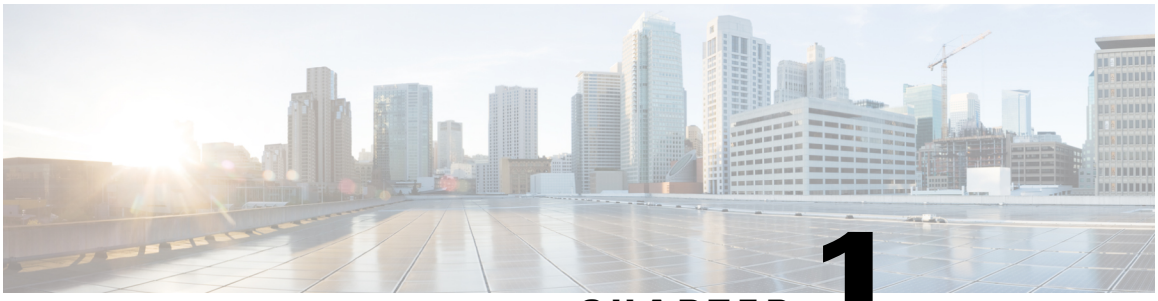
  https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.



CHAPTER 1

Using the Command-Line Interface

Information About Using the Command-Line Interface

Command Modes

The Cisco IOS user interface is divided into many different modes. The commands available to you depend on which mode you are currently in. Enter a question mark (?) at the system prompt to obtain a list of commands available for each command mode.

You can start a CLI session through a console connection, through Telnet, a SSH, or by using the browser.

When you start a session, you begin in user mode, often called user EXEC mode. Only a limited subset of the commands are available in user EXEC mode. For example, most of the user EXEC commands are one-time commands, such as **show** commands, which show the current configuration status, and **clear** commands, which clear counters or interfaces. The user EXEC commands are not saved when the switch reboots.

To have access to all commands, you must enter privileged EXEC mode. Normally, you must enter a password to enter privileged EXEC mode. From this mode, you can enter any privileged EXEC command or enter global configuration mode.

Using the configuration modes (global, interface, and line), you can make changes to the running configuration. If you save the configuration, these commands are stored and used when the switch reboots. To access the various configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and line configuration mode.

This table describes the main command modes, how to access each one, the prompt you see in that mode, and how to exit the mode.

Table 1: Command Mode Summary

Mode	Access Method	Prompt	Exit Method	About This Mode
User EXEC	Begin a session using Telnet, SSH, or console.	`Switch>`	Enter **logout** or **quit**.	Use this mode to • Change terminal settings. • Perform basic tests. • Display system information.
Privileged EXEC	While in user EXEC mode, enter the **enable** command.	`Switch#`	Enter **disable** to exit.	Use this mode to verify commands that you have entered. Use a password to protect access to this mode.
Global configuration	While in privileged EXEC mode, enter the **configure** command.	`Switch(config)#`	To exit to privileged EXEC mode, enter **exit** or **end**, or press **Ctrl-Z**.	Use this mode to configure parameters that apply to the entire switch.
VLAN configuration	While in global configuration mode, enter the **vlan** *vlan-id* command.	`Switch(config-vlan)#`	To exit to global configuration mode, enter the **exit** command. To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**.	Use this mode to configure VLAN parameters. When VTP mode is transparent, you can create extended-range VLANs (VLAN IDs greater than 1005) and save configurations in the switch startup configuration file.
Interface configuration	While in global configuration mode, enter the **interface** command (with a specific interface).	`Switch(config-if)#`	To exit to global configuration mode, enter **exit**. To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**.	Use this mode to configure parameters for the Ethernet ports.

Mode	Access Method	Prompt	Exit Method	About This Mode
Line configuration	While in global configuration mode, specify a line with the **line vty** or **line console** command.	`Switch(config-line)#`	To exit to global configuration mode, enter **exit**. To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**.	Use this mode to configure parameters for the terminal line.

Using the Help System

You can enter a question mark (?) at the system prompt to display a list of commands available for each command mode. You can also obtain a list of associated keywords and arguments for any command.

SUMMARY STEPS

1. **help**
2. *abbreviated-command-entry* **?**
3. *abbreviated-command-entry* <Tab>
4. **?**
5. *command* **?**
6. *command keyword* **?**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**help** **Example:** `Switch# help`	Obtains a brief description of the help system in any command mode.

Step 2	*abbreviated-command-entry* **?**
	Example:
	`Switch# di?`
	`dir disable disconnect`
		Obtains a list of commands that begin with a particular character string.

Step 3	*abbreviated-command-entry* <Tab>
	Example:
	`Switch# sh conf<tab>`
	`Switch# show configuration`
		Completes a partial command name.

	Command or Action	Purpose
Step 4	**?** **Example:** `Switch> ?`	Lists all commands available for a particular command mode.
Step 5	*command* **?** **Example:** `Switch> show ?`	Lists the associated keywords for a command.

Step 6	*command keyword* **?**
	
	Example:
	`Switch(config)# cdp holdtime ?`
	`  <10-255> Length of time (in sec) that receiver must keep this packet`
		Lists the associated arguments for a keyword.

Understanding Abbreviated Commands

You need to enter only enough characters for the switch to recognize the command as unique.

This example shows how to enter the **show configuration** privileged EXEC command in an abbreviated form:

```
Switch# show conf
```

No and Default Forms of Commands

Almost every configuration command also has a **no** form. In general, use the **no** form to disable a feature or function or reverse the action of a command. For example, the **no shutdown** interface configuration command reverses the shutdown of an interface. Use the command without the keyword **no** to reenable a disabled feature or to enable a feature that is disabled by default.

Configuration commands can also have a **default** form. The **default** form of a command returns the command setting to its default. Most commands are disabled by default, so the **default** form is the same as the **no** form. However, some commands are enabled by default and have variables set to certain default values. In these cases, the **default** command enables the command and sets variables to their default values.

CLI Error Messages

This table lists some error messages that you might encounter while using the CLI to configure your switch.

Table 2: Common CLI Error Messages

Error Message	Meaning	How to Get Help
`% Ambiguous command: "show con"`	You did not enter enough characters for your switch to recognize the command.	Reenter the command followed by a question mark (?) without any space between the command and the question mark. The possible keywords that you can enter with the command appear.
`% Incomplete command.`	You did not enter all of the keywords or values required by this command.	Reenter the command followed by a question mark (?) with a space between the command and the question mark. The possible keywords that you can enter with the command appear.
`% Invalid input detected at '^' marker.`	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all of the commands that are available in this command mode. The possible keywords that you can enter with the command appear.

Configuration Logging

You can log and view changes to the switch configuration. You can use the Configuration Change Logging and Notification feature to track changes on a per-session and per-user basis. The logger tracks each configuration command that is applied, the user who entered the command, the time that the command was entered, and the parser return code for the command. This feature includes a mechanism for asynchronous notification to registered applications whenever the configuration changes. You can choose to have the notifications sent to the syslog.

Note Only CLI or HTTP changes are logged.

How to Use the CLI to Configure Features

Configuring the Command History

The software provides a history or record of commands that you have entered. The command history feature is particularly useful for recalling long or complex commands or entries, including access lists. You can customize this feature to suit your needs.

Changing the Command History Buffer Size

By default, the switch records ten command lines in its history buffer. You can alter this number for a current terminal session or for all sessions on a particular line. This procedure is optional.

SUMMARY STEPS

1. **terminal history** [**size** *number-of-lines*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**terminal history** [**size** *number-of-lines*] **Example:** `Switch# terminal history size 200`	Changes the number of command lines that the switch records during the current terminal session in privileged EXEC mode. You can configure the size from 0 to 256.

Recalling Commands

To recall commands from the history buffer, perform one of the actions listed in this table. These actions are optional.

Note The arrow keys function only on ANSI-compatible terminals such as VT100s.

SUMMARY STEPS

1. **Ctrl-P** or use the **up arrow** key
2. **Ctrl-N** or use the **down arrow** key
3. **show history**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**Ctrl-P** or use the **up arrow** key	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Step 2	**Ctrl-N** or use the **down arrow** key	Returns to more recent commands in the history buffer after recalling commands with **Ctrl-P** or the up arrow key. Repeat the key sequence to recall successively more recent commands.

	Command or Action	Purpose
Step 3	**show history** **Example:** `Switch# show history`	Lists the last several commands that you just entered in privileged EXEC mode. The number of commands that appear is controlled by the setting of the **terminal history** global configuration command and the **history** line configuration command.

Disabling the Command History Feature

The command history feature is automatically enabled. You can disable it for the current terminal session or for the command line. This procedure is optional.

SUMMARY STEPS

1. **terminal no history**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**terminal no history** **Example:** `Switch# terminal no history`	Disables the feature during the current terminal session in privileged EXEC mode.

Enabling and Disabling Editing Features

Although enhanced editing mode is automatically enabled, you can disable it and reenable it.

SUMMARY STEPS

1. **terminal editing**
2. **terminal no editing**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**terminal editing** **Example:** `Switch# terminal editing`	Reenables the enhanced editing mode for the current terminal session in privileged EXEC mode.

	Command or Action	Purpose
Step 2	**terminal no editing** **Example:** `Switch# terminal no editing`	Disables the enhanced editing mode for the current terminal session in privileged EXEC mode.

Editing Commands Through Keystrokes

The keystrokes help you to edit the command lines. These keystrokes are optional.

Note The arrow keys function only on ANSI-compatible terminals such as VT100s.

Table 3: Editing Commands

Editing Commands	Description
Ctrl-B or use the **left arrow** key	Moves the cursor back one character.
Ctrl-F or use the **right arrow** key	Moves the cursor forward one character.
Ctrl-A	Moves the cursor to the beginning of the command line.
Ctrl-E	Moves the cursor to the end of the command line.
Esc B	Moves the cursor back one word.
Esc F	Moves the cursor forward one word.
Ctrl-T	Transposes the character to the left of the cursor with the character located at the cursor.
Delete or **Backspace** key	Erases the character to the left of the cursor.
Ctrl-D	Deletes the character at the cursor.
Ctrl-K	Deletes all characters from the cursor to the end of the command line.
Ctrl-U or **Ctrl-X**	Deletes all characters from the cursor to the beginning of the command line.
Ctrl-W	Deletes the word to the left of the cursor.

Esc D	Deletes from the cursor to the end of the word.
Esc C	Capitalizes at the cursor.
Esc L	Changes the word at the cursor to lowercase.
Esc U	Capitalizes letters from the cursor to the end of the word.
Ctrl-V or **Esc Q**	Designates a particular keystroke as an executable command, perhaps as a shortcut.
Return key	Scrolls down a line or screen on displays that are longer than the terminal screen can display. **Note** The More prompt is used for any output that has more lines than can be displayed on the terminal screen, including **show** command output. You can use the **Return** and **Space** bar keystrokes whenever you see the More prompt.
Space bar	Scrolls down one screen.
Ctrl-L or **Ctrl-R**	Redisplays the current command line if the switch suddenly sends a message to your screen.

Editing Command Lines That Wrap

You can use a wraparound feature for commands that extend beyond a single line on the screen. When the cursor reaches the right margin, the command line shifts ten spaces to the left. You cannot see the first ten characters of the line, but you can scroll back and check the syntax at the beginning of the command. The keystroke actions are optional.

To scroll back to the beginning of the command entry, press **Ctrl-B** or the left arrow key repeatedly. You can also press **Ctrl-A** to immediately move to the beginning of the line.

Note The arrow keys function only on ANSI-compatible terminals such as VT100s.

The following example shows how to wrap a command line that extends beyond a single line on the screen.

SUMMARY STEPS

1. **access-list**
2. **Ctrl-A**
3. **Return** key

DETAILED STEPS

	Command or Action	Purpose

Step 1	**access-list**
	
	Example:
	`Switch(config)# access-list 101 permit tcp 10.15.22.25 255.255.255.0 10.15.22.35`
	`Switch(config)# $ 101 permit tcp 10.15.22.25 255.255.255.0 10.15.22.35 255.25`
	`Switch(config)# $t tcp 10.15.22.25 255.255.255.0 131.108.1.20 255.255.255.0 eq`
	`Switch(config)# $15.22.25 255.255.255.0 10.15.22.35 255.255.255.0 eq 45`
		Displays the global configuration command entry that extends beyond one line.
		
		When the cursor first reaches the end of the line, the line is shifted ten spaces to the left and redisplayed. The dollar sign ($) shows that the line has been scrolled to the left. Each time the cursor reaches the end of the line, the line is again shifted ten spaces to the left.

Step 2	**Ctrl-A** **Example:** `Switch(config)# access-list 101 permit tcp 10.15.22.25 255.255.255.0 10.15.2$`	Checks the complete syntax. The dollar sign ($) appears at the end of the line to show that the line has been scrolled to the right.
Step 3	**Return** key	Execute the commands. The software assumes that you have a terminal screen that is 80 columns wide. If you have a different width, use the **terminal width** privileged EXEC command to set the width of your terminal. Use line wrapping with the command history feature to recall and modify previous complex command entries.

Searching and Filtering Output of show and more Commands

You can search and filter the output for **show** and **more** commands. This is useful when you need to sort through large amounts of output or if you want to exclude output that you do not need to see. Using these commands is optional.

SUMMARY STEPS

1. {**show** | **more**} *command* | {**begin** | **include** | **exclude**} *regular-expression*

DETAILED STEPS

	Command or Action	Purpose
Step 1	{**show** | **more**} *command* | {**begin** | **include** | **exclude**} *regular-expression*	Searches and filters the output.

	Command or Action	Purpose

	Example:
	`Switch# show interfaces | include protocol`
	`Vlan1 is up, line protocol is up`
	`Vlan10 is up, line protocol is down`
	`GigabitEthernet1/0/1 is up, line protocol is down`
	`GigabitEthernet1/0/2 is up, line protocol is up`
		Expressions are case sensitive. For example, if you enter **| exclude output**, the lines that contain **output** are not displayed, but the lines that contain **output** appear.

Accessing the CLI on a Switch Stack

You can access the CLI through a console connection, through Telnet, a SSH, or by using the browser.

You manage the switch stack and the stack member interfaces through the active switch. You cannot manage stack members on an individual switch basis. You can connect to the active switch through the console port or the Ethernet management port of one or more stack members. Be careful with using multiple CLI sessions on the active switch. Commands that you enter in one session are not displayed in the other sessions. Therefore, it is possible to lose track of the session from which you entered commands.

Note We recommend using one CLI session when managing the switch stack.

If you want to configure a specific stack member port, you must include the stack member number in the CLI command interface notation.

To debug the standby switch, use the **session standby ios** privileged EXEC command from the active switch to access the IOS console of the standby switch. To debug a specific stack member, use the **session switch** *stack-member-number* privileged EXEC command from the active switch to access the diagnostic shell of the stack member. For more information about these commands, see the switch command reference.

Accessing the CLI Through a Console Connection or Through Telnet

Before you can access the CLI, you must connect a terminal or a PC to the switch console or connect a PC to the Ethernet management port and then power on the switch, as described in the hardware installation guide that shipped with your switch.

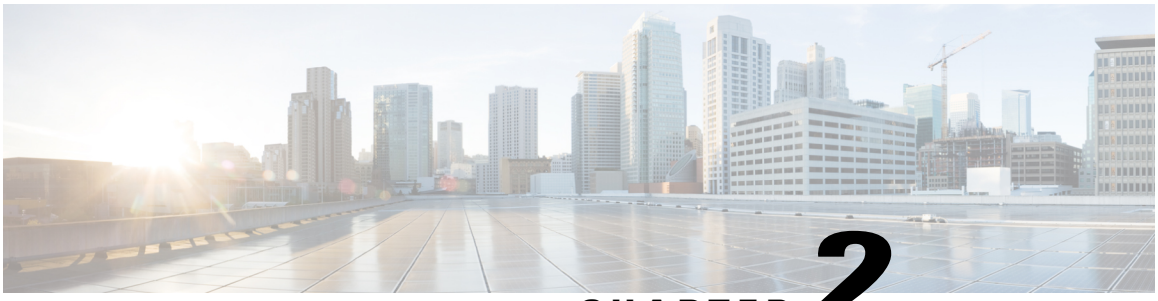
If your switch is already configured, you can access the CLI through a local console connection or through a remote Telnet session, but your switch must first be configured for this type of access.

You can use one of these methods to establish a connection with the switch:

- Connect the switch console port to a management station or dial-up modem, or connect the Ethernet management port to a PC. For information about connecting to the console or Ethernet management port, see the switch hardware installation guide.
- Use any Telnet TCP/IP or encrypted Secure Shell (SSH) package from a remote management station. The switch must have network connectivity with the Telnet or SSH client, and the switch must have an enable secret password configured.

- The switch supports up to 16 simultaneous Telnet sessions. Changes made by one Telnet user are reflected in all other Telnet sessions.
- The switch supports up to five simultaneous secure SSH sessions.

After you connect through the console port, through the Ethernet management port, through a Telnet session or through an SSH session, the user EXEC prompt appears on the management station.



CHAPTER 2

Configuring Cisco IOS Configuration Engine

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring the Configuration Engine

- Obtain the name of the configuration engine instance to which you are connecting.
- Because the CNS uses both the event bus and the configuration server to provide configurations to devices, you must define both ConfigID and Device ID for each configured switch.
- All switches configured with the **cns config partial** global configuration command must access the event bus. The DeviceID, as originated on the switch, must match the DeviceID of the corresponding switch definition in the Cisco Configuration Engine. You must know the hostname of the event bus to which you are connecting.

Related Topics

Cisco Networking Services IDs and Device Hostnames, on page 16
DeviceID, on page 17

Restrictions for Configuring the Configuration Engine

- Within the scope of a single instance of the configuration server, no two configured switches can share the same value for ConfigID.
- Within the scope of a single instance of the event bus, no two configured switches can share the same value for DeviceID.

Related Topics

Cisco Networking Services IDs and Device Hostnames, on page 16

Information About Configuring the Configuration Engine

Cisco Configuration Engine Software

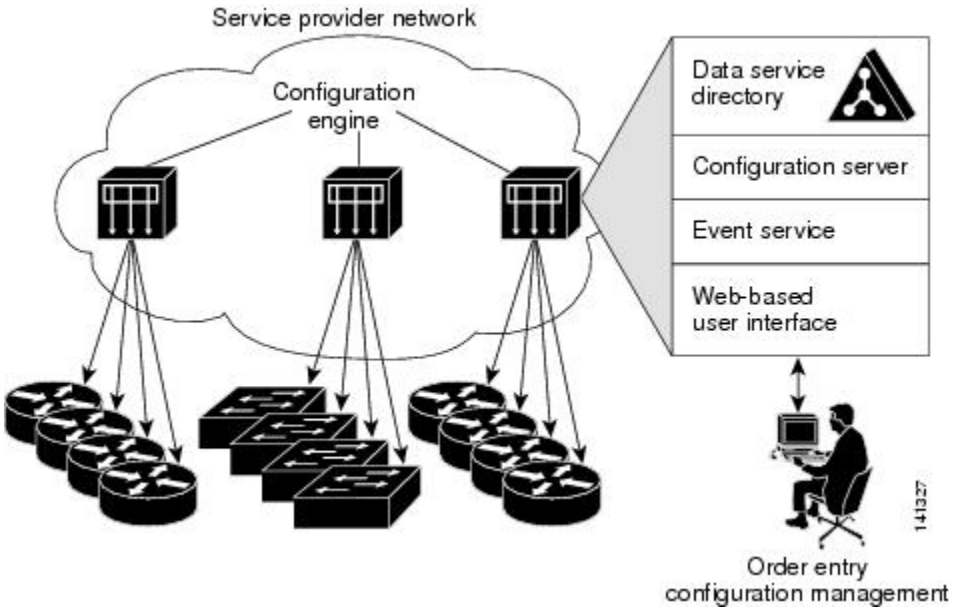
The Cisco Configuration Engine is network management utility software that acts as a configuration service for automating the deployment and management of network devices and services. Each Cisco Configuration Engine manages a group of Cisco devices (switches and routers) and the services that they deliver, storing their configurations and delivering them as needed. The Cisco Configuration Engine automates initial configurations and configuration updates by generating device-specific configuration changes, sending them to the device, executing the configuration change, and logging the results.

The Cisco Configuration Engine supports standalone and server modes and has these Cisco Networking Services (CNS) components:

- Configuration service:
  - Web server
  - File manager
  - Namespace mapping server
- Event service (event gateway)
- Data service directory (data models and schema)

In standalone mode, the Cisco Configuration Engine supports an embedded directory service. In this mode, no external directory or other data store is required. In server mode, the Cisco Configuration Engine supports the use of a user-defined external directory.

Figure 1: Cisco Configuration Engine Architectural Overview



Configuration Service

The Configuration Service is the core component of the Cisco Configuration Engine. It consists of a Configuration Server that works with Cisco IOS CNS agents on the switch. The Configuration Service delivers device and service configurations to the switch for initial configuration and mass reconfiguration by logical groups. Switches receive their initial configuration from the Configuration Service when they start up on the network for the first time.

The Configuration Service uses the CNS Event Service to send and receive configuration change events and to send success and failure notifications.

The Configuration Server is a web server that uses configuration templates and the device-specific configuration information stored in the embedded (standalone mode) or remote (server mode) directory.

Configuration templates are text files containing static configuration information in the form of CLI commands. In the templates, variables are specified by using Lightweight Directory Access Protocol (LDAP) URLs that reference the device-specific configuration information stored in a directory.

The Cisco IOS agent can perform a syntax check on received configuration files and publish events to show the success or failure of the syntax check. The configuration agent can either apply configurations immediately or delay the application until receipt of a synchronization event from the configuration server.

Event Service

The Cisco Configuration Engine uses the Event Service for receipt and generation of configuration events. The Event Service consists of an event agent and an event gateway. The event agent is on the switch and facilitates the communication between the switch and the event gateway on the Cisco Configuration Engine.

The Event Service is a highly capable publish-and-subscribe communication method. The Event Service uses subject-based addressing to send messages to their destinations. Subject-based addressing conventions define a simple, uniform namespace for messages and their destinations.

Related Topics

Enabling the CNS Event Agent, on page 20

NameSpace Mapper

The Cisco Configuration Engine includes the NameSpace Mapper (NSM) that provides a lookup service for managing logical groups of devices based on application, device or group ID, and event.

Cisco IOS devices recognize only event subject-names that match those configured in Cisco IOS software; for example, cisco.cns.config.load. You can use the namespace mapping service to designate events by using any desired naming convention. When you have populated your data store with your subject names, NSM changes your event subject-name strings to those known by Cisco IOS.

For a subscriber, when given a unique device ID and event, the namespace mapping service returns a set of events to which to subscribe. Similarly, for a publisher, when given a unique group ID, device ID, and event, the mapping service returns a set of events on which to publish.

Cisco Networking Services IDs and Device Hostnames

The Cisco Configuration Engine assumes that a unique identifier is associated with each configured switch. This unique identifier can take on multiple synonyms, where each synonym is unique within a particular namespace. The event service uses namespace content for subject-based addressing of messages.

The Cisco Configuration Engine intersects two namespaces, one for the event bus and the other for the configuration server. Within the scope of the configuration server namespace, the term *ConfigID* is the unique identifier for a device. Within the scope of the event bus namespace, the term *DeviceID* is the CNS unique identifier for a device.

Related Topics

Prerequisites for Configuring the Configuration Engine, on page 13
Restrictions for Configuring the Configuration Engine, on page 14

ConfigID

Each configured switch has a unique ConfigID, which serves as the key into the Cisco Configuration Engine directory for the corresponding set of switch CLI attributes. The ConfigID defined on the switch must match the ConfigID for the corresponding switch definition on the Cisco Configuration Engine.

The ConfigID is fixed at startup time and cannot be changed until the device restarts, even if the switch hostname is reconfigured.

DeviceID

Each configured switch participating on the event bus has a unique DeviceID, which is analogous to the switch source address so that the switch can be targeted as a specific destination on the bus.

The origin of the DeviceID is defined by the Cisco IOS hostname of the switch. However, the DeviceID variable and its usage reside within the event gateway adjacent to the switch.

The logical Cisco IOS termination point on the event bus is embedded in the event gateway, which in turn functions as a proxy on behalf of the switch. The event gateway represents the switch and its corresponding DeviceID to the event bus.

The switch declares its hostname to the event gateway immediately after the successful connection to the event gateway. The event gateway couples the DeviceID value to the Cisco IOS hostname each time this connection is established. The event gateway retains this DeviceID value for the duration of its connection to the switch.

Related Topics

Prerequisites for Configuring the Configuration Engine, on page 13

Hostname and DeviceID

The DeviceID is fixed at the time of the connection to the event gateway and does not change even when the switch hostname is reconfigured.

When changing the switch hostname on the switch, the only way to refresh the DeviceID is to break the connection between the switch and the event gateway. For instructions on refreshing DeviceIDs, see "Related Topics."

When the connection is reestablished, the switch sends its modified hostname to the event gateway. The event gateway redefines the DeviceID to the new value.

Caution When using the Cisco Configuration Engine user interface, you must first set the DeviceID field to the hostname value that the switch acquires *after*, not *before*, and you must reinitialize the configuration for your Cisco IOS CNS agent. Otherwise, subsequent partial configuration command operations may malfunction.

Related Topics

Refreshing DeviceIDs, on page 28

Hostname, DeviceID, and ConfigID

In standalone mode, when a hostname value is set for a switch, the configuration server uses the hostname as the DeviceID when an event is sent on hostname. If the hostname has not been set, the event is sent on the cn=<*value*> of the device.

In server mode, the hostname is not used. In this mode, the unique DeviceID attribute is always used for sending an event on the bus. If this attribute is not set, you cannot update the switch.

These and other associated attributes (tag value pairs) are set when you run **Setup** on the Cisco Configuration Engine.

Cisco IOS CNS Agents

The CNS event agent feature allows the switch to publish and subscribe to events on the event bus and works with the Cisco IOS CNS agent. These agents, embedded in the switch Cisco IOS software, allow the switch to be connected and automatically configured.

Initial Configuration

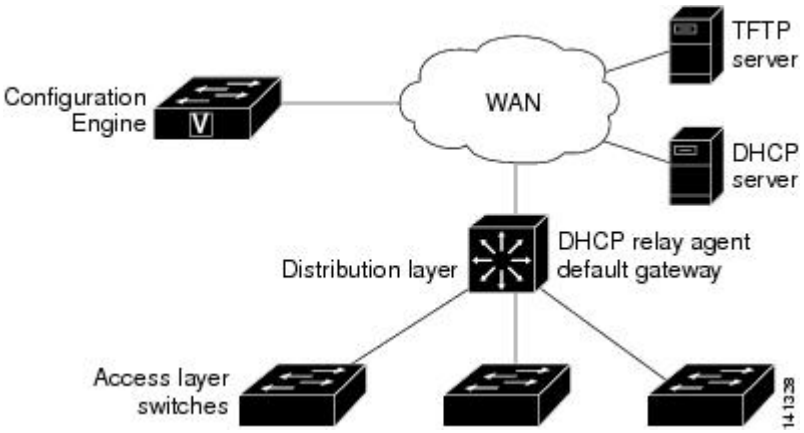
When the switch first comes up, it attempts to get an IP address by broadcasting a Dynamic Host Configuration Protocol (DHCP) request on the network. Assuming there is no DHCP server on the subnet, the distribution switch acts as a DHCP relay agent and forwards the request to the DHCP server. Upon receiving the request, the DHCP server assigns an IP address to the new switch and includes the Trivial File Transfer Protocol (TFTP) server Internet Protocol (IP) address, the path to the bootstrap configuration file, and the default gateway IP address in a unicast reply to the DHCP relay agent. The DHCP relay agent forwards the reply to the switch.

The switch automatically configures the assigned IP address on interface VLAN 1 (the default) and downloads the bootstrap configuration file from the TFTP server. Upon successful download of the bootstrap configuration file, the switch loads the file in its running configuration.

The Cisco IOS CNS agents initiate communication with the Configuration Engine by using the appropriate ConfigID and EventID. The Configuration Engine maps the Config ID to a template and downloads the full configuration file to the switch.

The following figure shows a sample network configuration for retrieving the initial bootstrap configuration file by using DHCP-based autoconfiguration.

Figure 2: Initial Configuration



Related Topics

Automated CNS Configuration, on page 19

Incremental (Partial) Configuration

After the network is running, new services can be added by using the Cisco IOS CNS agent. Incremental (partial) configurations can be sent to the switch. The actual configuration can be sent as an event payload by way of the event gateway (push operation) or as a signal event that triggers the switch to initiate a pull operation.

The switch can check the syntax of the configuration before applying it. If the syntax is correct, the switch applies the incremental configuration and publishes an event that signals success to the configuration server. If the switch does not apply the incremental configuration, it publishes an event showing an error status. When the switch has applied the incremental configuration, it can write it to nonvolatile random-access memory (NVRAM) or wait until signaled to do so.

Synchronized Configuration

When the switch receives a configuration, it can defer application of the configuration upon receipt of a write-signal event. The write-signal event tells the switch not to save the updated configuration into its NVRAM. The switch uses the updated configuration as its running configuration. This ensures that the switch configuration is synchronized with other network activities before saving the configuration in NVRAM for use at the next reboot.

Automated CNS Configuration

To enable automated CNS configuration of the switch, you must first complete the prerequisites listed in this topic. When you complete them, power on the switch. At the **setup** prompt, do nothing; the switch begins the initial configuration. When the full configuration file is loaded on your switch, you do not need to do anything else.

For more information on what happens during initial configuration, see "Related Topics."

Table 4: Prerequisites for Enabling Automatic Configuration

Device	Required Configuration
Access switch	Factory default (no configuration file)
Distribution switch	• IP helper address • Enable DHCP relay agent[1] • IP routing (if used as default gateway)
DHCP server	• IP address assignment • TFTP server IP address • Path to bootstrap configuration file on the TFTP server • Default gateway IP address

Device	Required Configuration
TFTP server	• A bootstrap configuration file that includes the CNS configuration commands that enable the switch to communicate with the Configuration Engine • The switch configured to use either the switch MAC address or the serial number (instead of the default hostname) to generate the ConfigID and EventID • The CNS event agent configured to push the configuration file to the switch
CNS Configuration Engine	One or more templates for each type of device, with the ConfigID of the device mapped to the template.

[1] A DHCP Relay is needed only when the DHCP Server is on a different subnet from the client.

Related Topics

Initial Configuration, on page 18

How to Configure the Configuration Engine

Enabling the CNS Event Agent

Note You must enable the CNS event agent on the switch before you enable the CNS configuration agent.

Beginning in privileged EXEC mode, follow these steps to enable the CNS event agent on the switch.

SUMMARY STEPS

1. **configure terminal**
2. **cns event** {*hostname* | *ip-address*} [*port-number*] [[**keepalive** *seconds retry-count*] [**failover-time** *seconds*] [**reconnect-time** *time*] | **backup**]
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**cns event** {*hostname* | *ip-address*} [*port-number*] [[**keepalive** *seconds retry-count*] [**failover-time** *seconds*] [**reconnect-time** *time*] | **backup**] **Example:** `Switch(config)# cns event 10.180.1.27 keepalive 120 10`	Enables the event agent, and enters the gateway parameters. • For {*hostname* | *ip-address*}, enter either the hostname or the IP address of the event gateway. • (Optional) For *port number*, enter the port number for the event gateway. The default port number is 11011. • (Optional) For **keepalive** *seconds*, enter how often the switch sends keepalive messages. For *retry-count*, enter the number of unanswered keepalive messages that the switch sends before the connection is terminated. The default for each is 0. • (Optional) For **failover-time** *seconds*, enter how long the switch waits for the primary gateway route after the route to the backup gateway is established. • (Optional) For **reconnect-time** *time*, enter the maximum time interval that the switch waits before trying to reconnect to the event gateway. • (Optional) Enter **backup** to show that this is the backup gateway. (If omitted, this is the primary gateway.) **Note** Though visible in the command-line help string, the **encrypt** and the **clock-timeout** *time* keywords are not supported.
Step 3	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

This example shows how to enable the CNS event agent, set the IP address gateway to 10.180.1.27, set 120 seconds as the keepalive interval, and set 10 as the retry count.

```
Switch(config)# cns event 10.180.1.27 keepalive 120 10
```

What to Do Next

To verify information about the event agent, use the **show cns event connections** command in privileged EXEC mode.

To disable the CNS event agent, use the **no cns event** { *ip-address* | *hostname* } global configuration command.

Related Topics

Event Service, on page 16

Enabling the Cisco IOS CNS Agent

Beginning in privileged EXEC mode, follow these steps to enable the Cisco IOS CNS agent on the switch.

Before You Begin

You must enable the CNS event agent on the switch before you enable this agent.

SUMMARY STEPS

1. **configure terminal**
2. **cns config initial** {*hostname* | *ip-address*} [*port-number*]
3. **cns config partial** {*hostname* | *ip-address*} [*port-number*]
4. **end**
5. Start the Cisco IOS CNS agent on the switch.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**cns config initial** {*hostname* | *ip-address*} [*port-number*] **Example:** `Switch(config)# cns config initial 10.180.1.27 10`	Enables the Cisco IOS CNS agent, and enters the configuration server parameters. • For {*hostname* | *ip-address*}, enter either the hostname or the IP address of the configuration server. • (Optional) For *port number*, enter the port number for the configuration server. This command enables the Cisco IOS CNS agent and initiates an initial configuration on the switch.
Step 3	**cns config partial** {*hostname* | *ip-address*} [*port-number*] **Example:** `Switch(config)# cns config partial 10.180.1.27 10`	Enables the Cisco IOS CNS agent, and enters the configuration server parameters. • For {*hostname* | *ip-address*}, enter either the hostname or the IP address of the configuration server. • (Optional) For *port number*, enter the port number for the configuration server.

	Command or Action	Purpose
		Enables the Cisco IOS CNS agent and initiates a partial configuration on the switch.
Step 4	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.
Step 5	Start the Cisco IOS CNS agent on the switch.	

What to Do Next

You can now use the Cisco Configuration Engine to remotely send incremental configurations to the switch.

Related Topics

Refreshing DeviceIDs, on page 28

Enabling an Initial Configuration for Cisco IOS CNS Agent

Beginning in privileged EXEC mode, follow these steps to enable the CNS configuration agent and initiate an initial configuration on the switch.

SUMMARY STEPS

1. **configure terminal**
2. **cns template connect** *name*
3. **cli** *config-text*
4. Repeat Steps 2 to 3 to configure another CNS connect template.
5. **exit**
6. **cns connect** *name* [**retries** *number*] [**retry-interval** *seconds*] [**sleep** *seconds*] [**timeout** *seconds*]
7. **discover** {**controller** *controller-type* | **dlci** [**subinterface** *subinterface-number*] | **interface** [*interface-type*] | **line** *line-type*}
8. **template** *name* [*... name*]
9. Repeat Steps 7 to 8 to specify more interface parameters and CNS connect templates in the CNS connect profile.
10. **exit**
11. **hostname** *name*
12. **ip route** *network-number*
13. **cns id** *interface num* {**dns-reverse** | **ipaddress** | **mac-address**} [**event**] [**image**]
14. **cns id** {**hardware-serial** | **hostname** | **string** *string* | **udi**} [**event**] [**image**]
15. **cns config initial** {*hostname* | *ip-address*} [*port-number*] [**event**] [**no-persist**] [**page** *page*] [**source** *ip-address*] [**syntax-check**]
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**cns template connect** *name* **Example:** `Switch(config)# cns template connect template-dhcp`	Enters CNS template connect configuration mode, and specifies the name of the CNS connect template.
Step 3	**cli** *config-text* **Example:** `Switch(config-tmpl-conn)# cli ip address dhcp`	Enters a command line for the CNS connect template. Repeat this step for each command line in the template.
Step 4	Repeat Steps 2 to 3 to configure another CNS connect template.	

	Command or Action	Purpose
Step 5	**exit** **Example:** `Switch(config)# exit`	Returns to global configuration mode.
Step 6	**cns connect** *name* [**retries** *number*] [**retry-interval** *seconds*] [**sleep** *seconds*] [**timeout** *seconds*] **Example:** `Switch(config)# cns connect dhcp`	Enters CNS connect configuration mode, specifies the name of the CNS connect profile, and defines the profile parameters. The switch uses the CNS connect profile to connect to the Configuration Engine. • Enter the *name* of the CNS connect profile. • (Optional) For **retries** *number*, enter the number of connection retries. The range is 1 to 30. The default is 3. • (Optional) For **retry-interval** *seconds*, enter the interval between successive connection attempts to the Configuration Engine. The range is 1 to 40 seconds. The default is 10 seconds. • (Optional) For **sleep** *seconds*, enter the amount of time before which the first connection attempt occurs. The range is 0 to 250 seconds. The default is 0. • (Optional) For **timeout** *seconds*, enter the amount of time after which the connection attempts end. The range is 10 to 2000 seconds. The default is 120.
Step 7	**discover** {**controller** *controller-type* | **dlci** [**subinterface** *subinterface-number*] | **interface** [*interface-type*] | **line** *line-type*} **Example:** `Switch(config-cns-conn)# discover interface gigabitethernet`	Specifies the interface parameters in the CNS connect profile. • For **controller** *controller-type*, enter the controller type. • For **dlci**, enter the active data-link connection identifiers (DLCIs). (Optional) For **subinterface** *subinterface-number*, specify the point-to-point subinterface number that is used to search for active DLCIs. • For **interface** [*interface-type*], enter the type of interface. • For **line** *line-type*, enter the line type.
Step 8	**template** *name* [*... name*] **Example:** `Switch(config-cns-conn)# template template-dhcp`	Specifies the list of CNS connect templates in the CNS connect profile to be applied to the switch configuration. You can specify more than one template.
Step 9	Repeat Steps 7 to 8 to specify more interface parameters and CNS connect templates in the CNS connect profile.	

	Command or Action	Purpose
Step 10	**exit** **Example:** `Switch(config-cns-conn)# exit`	Returns to global configuration mode.
Step 11	**hostname** *name* **Example:** `Switch(config)# hostname device1`	Enters the hostname for the switch.
Step 12	**ip route** *network-number* **Example:** `RemoteSwitch(config)# ip route 172.28.129.22 255.255.255.255 11.11.11.1`	(Optional) Establishes a static route to the Configuration Engine whose IP address is *network-number*.
Step 13	**cns id** *interface num* {**dns-reverse** | **ipaddress** | **mac-address**} [**event**] [**image**] **Example:** `RemoteSwitch(config)# cns id GigabitEthernet1/0/1 ipaddress`	(Optional) Sets the unique EventID or ConfigID used by the Configuration Engine. If you enter this command, do not enter the **cns id** {**hardware-serial** | **hostname** | **string** *string* | **udi**} [**event**] [**image**] command. • For *interface num*, enter the type of interface. For example, ethernet, group-async, loopback, or virtual-template. This setting specifies from which interface the IP or MAC address should be retrieved to define the unique ID. • For {**dns-reverse** | **ipaddress** | **mac-address**}, enter **dns-reverse** to retrieve the hostname and assign it as the unique ID, enter **ipaddress** to use the IP address, or enter **mac-address** to use the MAC address as the unique ID. • (Optional) Enter **event** to set the ID to be the event-id value used to identify the switch. • (Optional) Enter **image** to set the ID to be the image-id value used to identify the switch. **Note** If both the **event** and **image** keywords are omitted, the image-id value is used to identify the switch.
Step 14	**cns id** {**hardware-serial** | **hostname** | **string** *string* | **udi**} [**event**] [**image**] **Example:** `RemoteSwitch(config)# cns id hostname`	(Optional) Sets the unique EventID or ConfigID used by the Configuration Engine. If you enter this command, do not enter the **cns id** *interface num* {**dns-reverse** | **ipaddress** | **mac-address**} [**event**] [**image**] command. • For { **hardware-serial** | **hostname** | **string** *string* | **udi** }, enter **hardware-serial** to set the switch serial number as the unique ID, enter **hostname** (the default) to select the switch hostname as the unique ID, enter an arbitrary text string for **string** *string* as the unique ID, or enter **udi** to set the unique device identifier (UDI) as the unique ID.

	Command or Action	Purpose
Step 15	**cns config initial** {*hostname* | *ip-address*} [*port-number*] [**event**] [**no-persist**] [**page** *page*] [**source** *ip-address*] [**syntax-check**] **Example:** `RemoteSwitch(config)# cns config initial 10.1.1.1 no-persist`	Enables the Cisco IOS agent, and initiates an initial configuration. • For {*hostname* | *ip-address*}, enter the hostname or the IP address of the configuration server. • (Optional) For *port-number*, enter the port number of the configuration server. The default port number is 80. • (Optional) Enable **event** for configuration success, failure, or warning messages when the configuration is finished. • (Optional) Enable **no-persist** to suppress the automatic writing to NVRAM of the configuration pulled as a result of entering the **cns config initial** global configuration command. If the **no-persist** keyword is not entered, using the **cns config initial** command causes the resultant configuration to be automatically written to NVRAM. • (Optional) For **page** *page*, enter the web page of the initial configuration. The default is /Config/config/asp. • (Optional) Enter **source** *ip-address* to use for source IP address. • (Optional) Enable **syntax-check** to check the syntax when this parameter is entered. **Note** Though visible in the command-line help string, the **encrypt**, **status** *url*, and **inventory** keywords are not supported.
Step 16	**end** **Example:** `RemoteSwitch(config)# end`	Returns to privileged EXEC mode.

This example shows how to configure an initial configuration on a remote switch when the switch configuration is unknown (the CNS Zero Touch feature).

```
Switch(config)# cns template connect template-dhcp
Switch(config-tmpl-conn)# cli ip address dhcp
Switch(config-tmpl-conn)# exit
Switch(config)# cns template connect ip-route
Switch(config-tmpl-conn)# cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
Switch(config-tmpl-conn)# exit
Switch(config)# cns connect dhcp
Switch(config-cns-conn)# discover interface gigabitethernet
Switch(config-cns-conn)# template template-dhcp
Switch(config-cns-conn)# template ip-route
Switch(config-cns-conn)# exit
Switch(config)# hostname RemoteSwitch
RemoteSwitch(config)# cns config initial 10.1.1.1 no-persist
```

This example shows how to configure an initial configuration on a remote switch when the switch IP address is known. The Configuration Engine IP address is 172.28.129.22.

```
Switch(config)# cns template connect template-dhcp
Switch(config-tmpl-conn)# cli ip address dhcp
Switch(config-tmpl-conn)# exit
Switch(config)# cns template connect ip-route
Switch(config-tmpl-conn)# cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
Switch(config-tmpl-conn)# exit
Switch(config)# cns connect dhcp
Switch(config-cns-conn)# discover interface gigabitethernet
Switch(config-cns-conn)# template template-dhcp
Switch(config-cns-conn)# template ip-route
Switch(config-cns-conn)# exit
Switch(config)# hostname RemoteSwitch
RemoteSwitch(config)# ip route 172.28.129.22 255.255.255.255 11.11.11.1
RemoteSwitch(config)# cns id ethernet 0 ipaddress
RemoteSwitch(config)# cns config initial 172.28.129.22 no-persist
```

What to Do Next

To verify information about the configuration agent, use the **show cns config connections** command in privileged EXEC mode.

To disable the CNS Cisco IOS agent, use the **no cns config initial** { *ip-address* | *hostname* } global configuration command.

Refreshing DeviceIDs

Beginning in privileged EXEC mode, follow these steps to refresh a DeviceID when changing the hostname on the switch.

SUMMARY STEPS

1. **show cns config connections**
2. Make sure that the CNS event agent is properly connected to the event gateway.
3. **show cns event connections**
4. Record from the output of Step 3 the information for the currently connected connection listed below. You will be using the IP address and port number in subsequent steps of these instructions.
5. **configure terminal**
6. **no cns event** *ip-address port-number*
7. **cns event** *ip-address port-number*
8. **end**
9. Make sure that you have reestablished the connection between the switch and the event connection by examining the output from **show cns event connections**.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show cns config connections** **Example:** `Switch# show cns config connections`	Displays whether the CNS event agent is connecting to the gateway, connected, or active, and the gateway used by the event agent, its IP address and port number.
Step 2	Make sure that the CNS event agent is properly connected to the event gateway.	Examine the output of **show cns config connections** for the following: • Connection is active. • Connection is using the currently configured switch hostname. The DeviceID will be refreshed to correspond to the new hostname configuration using these instructions.
Step 3	**show cns event connections** **Example:** `Switch# show cns event connections`	Displays the event connection information for your switch.
Step 4	Record from the output of Step 3 the information for the currently connected connection listed below. You will be using the IP address and port number in subsequent steps of these instructions.	
Step 5	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 6	**no cns event** *ip-address port-number* **Example:** `Switch(config)# no cns event 172.28.129.22 2012`	Specifies the IP address and port number that you recorded in Step 4 in this command. This command breaks the connection between the switch and the event gateway. It is necessary to first break, then reestablish, this connection to refresh the DeviceID.
Step 7	**cns event** *ip-address port-number* **Example:** `Switch(config)# cns event 172.28.129.22 2012`	Specifies the IP address and port number that you recorded in Step 4 in this command. This command reestablishes the connection between the switch and the event gateway.
Step 8	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 9	Make sure that you have reestablished the connection between the switch and the event connection by examining the output from **show cns event connections**.	

Related Topics

Enabling the Cisco IOS CNS Agent, on page 22
Hostname and DeviceID, on page 17

Enabling a Partial Configuration for Cisco IOS CNS Agent

Beginning in privileged EXEC mode, follow these steps to enable the Cisco IOS CNS agent and to initiate a partial configuration on the switch.

SUMMARY STEPS

1. **configure terminal**
2. **cns config partial** {*ip-address* | *hostname*} [*port-number*] [**source** *ip-address*]
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**cns config partial** {*ip-address* | *hostname*} [*port-number*] [**source** *ip-address*] **Example:** `Switch(config)# cns config partial 172.28.129.22 2013`	Enables the configuration agent, and initiates a partial configuration. • For {*ip-address* | *hostname*}, enter the IP address or the hostname of the configuration server. • (Optional) For *port-number*, enter the port number of the configuration server. The default port number is 80. • (Optional) Enter **source** *ip-address* to use for the source IP address. **Note** Though visible in the command-line help string, the **encrypt** keyword is not supported.

	Command or Action	Purpose
Step 3	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

What to Do Next

To verify information about the configuration agent, use either the **show cns config stats** or the **show cns config outstanding** command in privileged EXEC mode.

To disable the Cisco IOS agent, use the **no cns config partial** { *ip-address* | *hostname* } global configuration command. To cancel a partial configuration, use the **cns config cancel** global configuration command.

Monitoring CNS Configurations

Table 5: CNS show Commands

Command	Purpose
show cns config connections `Switch# show cns config connections`	Displays the status of the CNS Cisco IOS CNS agent connections.
show cns config outstanding `Switch# show cns config outstanding`	Displays information about incremental (partial) CNS configurations that have started but are not yet completed.
show cns config stats `Switch# show cns config stats`	Displays statistics about the Cisco IOS CNS agent.
show cns event connections `Switch# show cns event connections`	Displays the status of the CNS event agent connections.
show cns event gateway `Switch# show cns event gateway`	Displays the event gateway information for your switch.
show cns event stats `Switch# show cns event stats`	Displays statistics about the CNS event agent.
show cns event subject `Switch# show cns event subject`	Displays a list of event agent subjects that are subscribed to by applications.

Additional References

Related Documents

Related Topic	Document Title
Configuration Engine Setup	*Cisco Configuration Engine Installation and Setup Guide, 1.5 for Linux* http://www.cisco.com/en/US/docs/net_mgmt/configuration_engine/1.5/installation_linux/guide/setup_1.html

Error Message Decoder

Description	Link
To help you research and resolve system error messages in this release, use the Error Message Decoder tool.	https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

MIBs

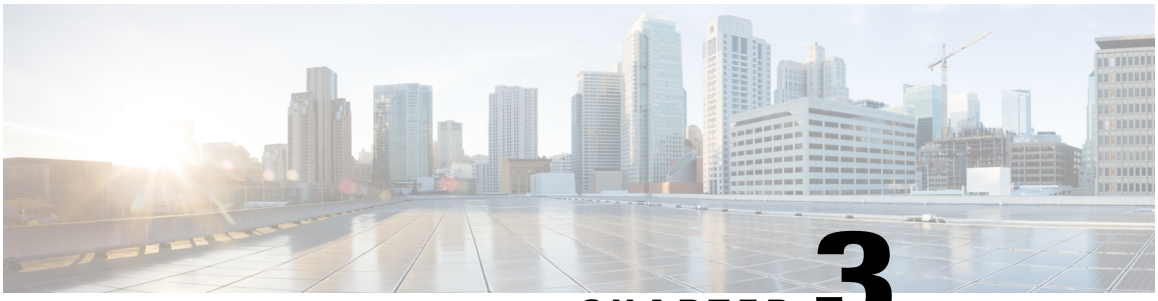
MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History and Information for the Configuration Engine

Release	Modification
Cisco IOS XE 3.3SE	This feature was introduced.



CHAPTER 3

Configuring the Cisco Discovery Protocol

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About CDP

CDP Overview

CDP is a device discovery protocol that runs over Layer 2 (the data-link layer) on all Cisco-manufactured devices (routers, bridges, access servers, controllers, and switches) and allows network management applications to discover Cisco devices that are neighbors of already known devices. With CDP, network management applications can learn the device type and the Simple Network Management Protocol (SNMP) agent address of neighboring devices running lower-layer, transparent protocols. This feature enables applications to send SNMP queries to neighboring devices.

CDP runs on all media that support Subnetwork Access Protocol (SNAP). Because CDP runs over the data-link layer only, two systems that support different network-layer protocols can learn about each other.

Each CDP-configured device sends periodic messages to a multicast address, advertising at least one address at which it can receive SNMP messages. The advertisements also contain time-to-live, or holdtime information, which is the length of time a receiving device holds CDP information before discarding it. Each device also listens to the messages sent by other devices to learn about neighboring devices.

On the switch, CDP enables Network Assistant to display a graphical view of the network. The switch uses CDP to find cluster candidates and maintain information about cluster members and other devices up to three cluster-enabled devices away from the command switch by default.

CDP and Stacks

A switch stack appears as a single switch in the network. Therefore, CDP discovers the switch stack, not the individual stack members. The switch stack sends CDP messages to neighboring network devices when there are changes to the switch stack membership, such as stack members being added or removed.

Default CDP Configuration

This table shows the default CDP configuration.

Feature	Default Setting
CDP global state	Enabled
CDP interface state	Enabled
CDP timer (packet update frequency)	60 seconds
CDP holdtime (before discarding)	180 seconds
CDP Version-2 advertisements	Enabled

How to Configure CDP

Configuring CDP Characteristics

You can configure these CDP characteristics:

- Frequency of CDP updates
- Amount of time to hold the information before discarding it
- Whether or not to send Version-2 advertisements

Note Steps 2 through 4 are all optional and can be performed in any order.

Beginning in privileged EXEC mode, follow these steps to configure these characteristics.

SUMMARY STEPS

1. **configure terminal**
2. **cdp timer** *seconds*
3. **cdp holdtime** *seconds*
4. **cdp advertise-v2**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**cdp timer** *seconds* **Example:** `Switch(config)# cdp timer 20`	(Optional) Sets the transmission frequency of CDP updates in seconds. The range is 5 to 254; the default is 60 seconds.
Step 3	**cdp holdtime** *seconds* **Example:** `Switch(config)# cdp holdtime 60`	(Optional) Specifies the amount of time a receiving device should hold the information sent by your device before discarding it. The range is 10 to 255 seconds; the default is 180 seconds.
Step 4	**cdp advertise-v2** **Example:** `Switch(config)# cdp advertise-v2`	(Optional) Configures CDP to send Version-2 advertisements. This is the default state.
Step 5	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Example

The following example shows how to configure CDP characteristics:

```
Switch# configure terminal
Switch(config)# cdp timer 50
Switch(config)# cdp holdtime 120
Switch(config)# cdp advertise-v2
Switch(config)# end
```

What to Do Next

Use the **no** form of the CDP commands to return to the default settings.

Related Topics

Monitoring and Maintaining CDP, on page 42

Disabling CDP

CDP is enabled by default.

Note Switch clusters and other Cisco devices (such as Cisco IP Phones) regularly exchange CDP messages. Disabling CDP can interrupt cluster discovery and device connectivity.

Beginning in privileged EXEC mode, follow these steps to disable the CDP device discovery capability.

SUMMARY STEPS

1. **configure terminal**
2. **no cdp run**
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**no cdp run** **Example:** `Switch(config)# no cdp run`	Disables CDP.
Step 3	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

What to Do Next

You must reenable CDP to use it.

Related Topics

Enabling CDP, on page 39

Enabling CDP

CDP is enabled by default.

Note Switch clusters and other Cisco devices (such as Cisco IP Phones) regularly exchange CDP messages. Disabling CDP can interrupt cluster discovery and device connectivity.

Beginning in privileged EXEC mode, follow these steps to enable CDP when it has been disabled.

Before You Begin

CDP must be disabled, or it cannot be enabled.

SUMMARY STEPS

1. **configure terminal**
2. **cdp run**
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**cdp run** **Example:** `Switch(config)# cdp run`	Enables CDP if it has been disabled.
Step 3	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Example

The following example shows how to enable CDP if it has been disabled:

```
Switch# configure terminal
Switch(config)# cdp run
Switch(config)# end
```

What to Do Next

Use the **show run all** command to show that CDP has been enabled. If you enter only **show run**, the enabling of CDP may not be displayed.

Related Topics

Disabling CDP, on page 38

Disabling CDP on an Interface

CDP is enabled by default on all supported interfaces to send and to receive CDP information.

Note Switch clusters and other Cisco devices (such as Cisco IP Phones) regularly exchange CDP messages. Disabling CDP can interrupt cluster discovery and device connectivity.

Beginning in privileged EXEC mode, follow these steps to disable CDP on a port.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **no cdp enable**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Switch(config)# interface gigabitethernet1/0/1`	Specifies the interface on which you are disabling CDP, and enters interface configuration mode.

	Command or Action	Purpose
Step 3	**no cdp enable** **Example:** `Switch(config-if)# no cdp enable`	Disables CDP on the interface specified in Step 2.
Step 4	**end** **Example:** `Switch(config-if)# end`	Returns to privileged EXEC mode.

Related Topics

Enabling CDP on an Interface, on page 41

Enabling CDP on an Interface

CDP is enabled by default on all supported interfaces to send and to receive CDP information.

Note Switch clusters and other Cisco devices (such as Cisco IP Phones) regularly exchange CDP messages. Disabling CDP can interrupt cluster discovery and device connectivity.

Beginning in privileged EXEC mode, follow these steps to enable CDP on a port on which it has been disabled.

Before You Begin

CDP must be disabled on the port that you are trying to CDP enable on, or it cannot be enabled.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **cdp enable**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.

	Command or Action	Purpose
Step 2	**interface** *interface-id* **Example:** `Switch(config)# interface gigabitethernet1/0/1`	Specifies the interface on which you are enabling CDP, and enters interface configuration mode.
Step 3	**cdp enable** **Example:** `Switch(config-if)# cdp enable`	Enables CDP on a disabled interface.
Step 4	**end** **Example:** `Switch(config-if)# end`	Returns to privileged EXEC mode.

Example

The following example shows how to enable CDP on a disabled port:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# cdp enable
Switch(config-if)# end
```

Related Topics

Disabling CDP on an Interface, on page 40

Monitoring and Maintaining CDP

Table 6: Commands for Displaying CDP Information

Command	Description
clear cdp counters	Resets the traffic counters to zero.
clear cdp table	Deletes the CDP table of information about neighbors.
show cdp	Displays global information, such as frequency of transmissions and the holdtime for packets being sent.

Command	Description
show cdp entry *entry-name* [**version**] [**protocol**]	Displays information about a specific neighbor. You can enter an asterisk (*) to display all CDP neighbors, or you can enter the name of the neighbor about which you want information. You can also limit the display to information about the protocols enabled on the specified neighbor or information about the version of software running on the device.
show cdp interface [*interface-id*]	Displays information about interfaces where CDP is enabled. You can limit the display to the interface about which you want information.
show cdp neighbors [*interface-id*] [*detail*]	Displays information about neighbors, including device type, interface type and number, holdtime settings, capabilities, platform, and port ID. You can limit the display to neighbors of a specific interface or expand the display to provide more detailed information.
show cdp traffic	Displays CDP counters, including the number of packets sent and received and checksum errors.

Related Topics

Configuring CDP Characteristics, on page 36

Additional References

Related Documents

Related Topic	Document Title
System Management Commands	*Configuration Fundamentals Command Reference, Cisco IOS XE Release 3S (Catalyst 3850 Switches)*

Error Message Decoder

Description	Link
To help you research and resolve system error messages in this release, use the Error Message Decoder tool.	https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

MIBs

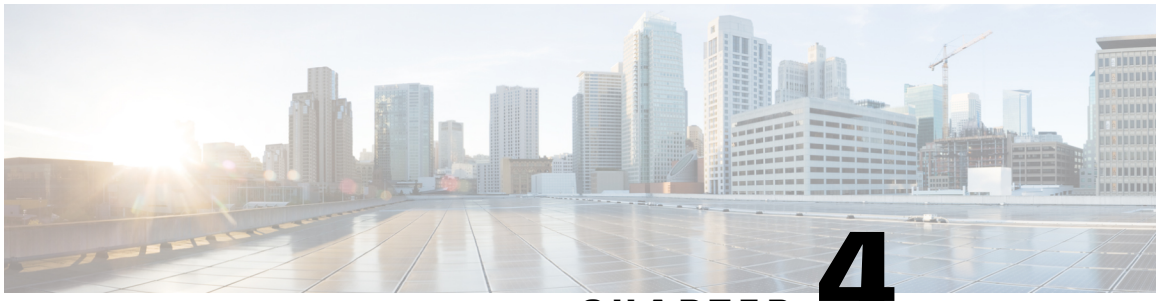
MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History and Information for Cisco Discovery Protocol

Release	Modification
Cisco IOS XE 3.3SE	This feature was introduced.



CHAPTER 4

Configuring Simple Network Management Protocol

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SNMP

Supported SNMP Versions

This software release supports the following SNMP versions:

- SNMPv1—The Simple Network Management Protocol, a Full Internet Standard, defined in RFC 1157.
- SNMPv2C replaces the Party-based Administrative and Security Framework of SNMPv2Classic with the community-string-based Administrative Framework of SNMPv2C while retaining the bulk retrieval and improved error handling of SNMPv2Classic. It has these features:

- SNMPv2—Version 2 of the Simple Network Management Protocol, a Draft Internet Standard, defined in RFCs 1902 through 1907.
- SNMPv2C—The community-string-based Administrative Framework for SNMPv2, an Experimental Internet Protocol defined in RFC 1901.

- SNMPv3—Version 3 of the SNMP is an interoperable standards-based protocol defined in RFCs 2273 to 2275. SNMPv3 provides secure access to devices by authenticating and encrypting packets over the network and includes these security features:
  - Message integrity—Ensures that a packet was not tampered with in transit.
  - Authentication—Determines that the message is from a valid source.
  - Encryption—Mixes the contents of a package to prevent it from being read by an unauthorized source.

Note To select encryption, enter the **priv** keyword.

Both SNMPv1 and SNMPv2C use a community-based form of security. The community of managers able to access the agent's MIB is defined by an IP address access control list and password.

SNMPv2C includes a bulk retrieval function and more detailed error message reporting to management stations. The bulk retrieval function retrieves tables and large quantities of information, minimizing the number of round-trips required. The SNMPv2C improved error-handling includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1. Error return codes in SNMPv2C report the error type.

SNMPv3 provides for both security models and security levels. A security model is an authentication strategy set up for a user and the group within which the user resides. A security level is the permitted level of security within a security model. A combination of the security level and the security model determine which security method is used when handling an SNMP packet. Available security models are SNMPv1, SNMPv2C, and SNMPv3.

The following table identifies characteristics and compares different combinations of security models and levels:

Table 7: SNMP Security Models and Levels

Model	Level	Authentication	Encryption	Result
SNMPv1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
SNMPv2C	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
SNMPv3	noAuthNoPriv	Username	No	Uses a username match for authentication.

Model	Level	Authentication	Encryption	Result
SNMPv3	authNoPriv	Message Digest 5 (MD5) or Secure Hash Algorithm (SHA)	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms.
SNMPv3	authPriv	MD5 or SHA	Data Encryption Standard (DES) or Advanced Encryption Standard (AES)	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Allows specifying the User-based Security Model (USM) with these encryption algorithms: • DES 56-bit encryption in addition to authentication based on the CBC-DES (DES-56) standard. • 3DES 168-bit encryption • AES 128-bit, 192-bit, or 256-bit encryption

You must configure the SNMP agent to use the SNMP version supported by the management station. Because an agent can communicate with multiple managers, you can configure the software to support communications using SNMPv1, SNMPv2C, or SNMPv3.

Restrictions for SNMP

Version Restrictions

- SNMPv1 does not support informs.

Information About SNMP

SNMP Overview

SNMP is an application-layer protocol that provides a message format for communication between managers and agents. The SNMP system consists of an SNMP manager, an SNMP agent, and a management information base (MIB). The SNMP manager can be part of a network management system (NMS) such as Cisco Prime Infrastructure. The agent and MIB reside on the switch. To configure SNMP on the switch, you define the relationship between the manager and the agent.

The SNMP agent contains MIB variables whose values the SNMP manager can request or change. A manager can get a value from an agent or store a value into the agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to a manager's requests to get or set data.

An agent can send unsolicited traps to the manager. Traps are messages alerting the SNMP manager to a condition on the network. Traps can mean improper user authentication, restarts, link status (up or down), MAC address tracking, closing of a TCP connection, loss of connection to a neighbor, or other significant events.

The active switch handles the SNMP requests and traps for the whole switch stack. The active switch transparently manages any requests or traps that are related to all stack members. When a new active switch is elected, the new active switch continues to handle SNMP requests and traps as configured on the previous active switch, assuming that IP connectivity to the SNMP management stations is still in place after the new active switch has taken control.

SNMP Manager Functions

The SNMP manager uses information in the MIB to perform the operations described in the following table:

Table 8: SNMP Operations

Operation	Description
get-request	Retrieves a value from a specific variable.
get-next-request	Retrieves a value from a variable within a table.[2]
get-bulk-request[3]	Retrieves large blocks of data, such as multiple rows in a table, that would otherwise require the transmission of many small blocks of data.
get-response	Replies to a get-request, get-next-request, and set-request sent by an NMS.
set-request	Stores a value in a specific variable.
trap	An unsolicited message sent by an SNMP agent to an SNMP manager when some event has occurred.

2 With this operation, an SNMP manager does not need to know the exact variable name. A sequential search is performed to find the needed variable from within a table.
3 The get-bulk command only works with SNMPv2 or later.

SNMP Agent Functions

The SNMP agent responds to SNMP manager requests as follows:

- Get a MIB variable—The SNMP agent begins this function in response to a request from the NMS. The agent retrieves the value of the requested MIB variable and responds to the NMS with that value.
- Set a MIB variable—The SNMP agent begins this function in response to a message from the NMS. The SNMP agent changes the value of the MIB variable to the value requested by the NMS.

The SNMP agent also sends unsolicited trap messages to notify an NMS that a significant event has occurred on the agent. Examples of trap conditions include, but are not limited to, when a port or module goes up or down, when spanning-tree topology changes occur, and when authentication failures occur.

SNMP Community Strings

SNMP community strings authenticate access to MIB objects and function as embedded passwords. In order for the NMS to access the switch, the community string definitions on the NMS must match at least one of the three community string definitions on the switch.

A community string can have one of the following attributes:

- Read-only (RO)—Gives all objects in the MIB except the community strings read access to authorized management stations, but does not allow write access.
- Read-write (RW)—Gives all objects in the MIB read and write access to authorized management stations, but does not allow access to the community strings.
- When a cluster is created, the command switch manages the exchange of messages among member switches and the SNMP application. The Network Assistant software appends the member switch number (@esN, where N is the switch number) to the first configured RW and RO community strings on the command switch and propagates them to the member switches.

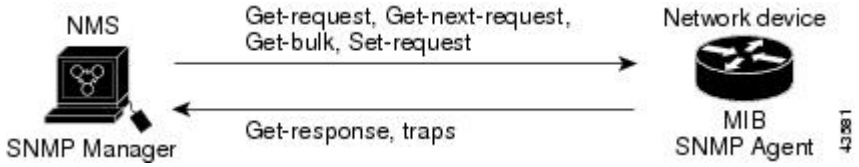
SNMP MIB Variables Access

An example of an NMS is the Cisco Prime Infrastructure network management software. Cisco Prime Infrastructure 2.0 software uses the switch MIB variables to set device variables and to poll devices on the network for specific information. The results of a poll can be displayed as a graph and analyzed to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, monitor traffic loads, and more.

As shown in the figure, the SNMP agent gathers data from the MIB. The agent can send traps, or notification of certain events, to the SNMP manager, which receives and processes the traps. Traps alert the SNMP manager to a condition on the network such as improper user authentication, restarts, link status (up or down), MAC

address tracking, and so forth. The SNMP agent also responds to MIB-related queries sent by the SNMP manager in get-request, get-next-request, and set-request format.

Figure 3: SNMP Network



SNMP Notifications

SNMP allows the switch to send notifications to SNMP managers when particular events occur. SNMP notifications can be sent as traps or inform requests. In command syntax, unless there is an option in the command to select either traps or informs, the keyword traps refers to either traps or informs, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs.

Note SNMPv1 does not support informs.

Traps are unreliable because the receiver does not send an acknowledgment when it receives a trap, and the sender cannot determine if the trap was received. When an SNMP manager receives an inform request, it acknowledges the message with an SNMP response protocol data unit (PDU). If the sender does not receive a response, the inform request can be sent again. Because they can be resent, informs are more likely than traps to reach their intended destination.

The characteristics that make informs more reliable than traps also consume more resources in the switch and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request is held in memory until a response is received or the request times out. Traps are sent only once, but an inform might be resent or retried several times. The retries increase traffic and contribute to a higher overhead on the network. Therefore, traps and informs require a trade-off between reliability and resources. If it is important that the SNMP manager receive every notification, use inform requests. If traffic on the network or memory in the switch is a concern and notification is not required, use traps.

SNMP ifIndex MIB Object Values

In an NMS, the IF-MIB generates and assigns an interface index (ifIndex) object value that is a unique number greater than zero to identify a physical or a logical interface. When the switch reboots or the switch software is upgraded, the switch uses this same value for the interface. For example, if the switch assigns a port 2 an ifIndex value of 10003, this value is the same after the switch reboots.

The switch uses one of the values in the following table to assign an ifIndex value to an interface:

Table 9: ifIndex Values

Interface Type	ifIndex Range
SVI[4]	1–4999

Interface Type	ifIndex Range
EtherChannel	5000–5012
Loopback	5013–5077
Tunnel	5078–5142
Physical (such as Gigabit Ethernet or SFP[5]-module interfaces)	10000–14500
Null	14501

4 SVI = switch virtual interface
5 SFP = small form-factor pluggable

Default SNMP Configuration

Feature	Default Setting
SNMP agent	Disabled[6].
SNMP trap receiver	None configured.
SNMP traps	None enabled except the trap for TCP connections (tty).
SNMP version	If no version keyword is present, the default is Version 1.
SNMPv3 authentication	If no keyword is entered, the default is the **noauth** (noAuthNoPriv) security level.
SNMP notification type	If no type is specified, all notifications are sent.

6 This is the default when the switch starts and the startup configuration does not have any **snmp-server** global configuration commands.

SNMP Configuration Guidelines

If the switch starts and the switch startup configuration has at least one **snmp-server** global configuration command, the SNMP agent is enabled.

An SNMP *group* is a table that maps SNMP users to SNMP views. An SNMP *user* is a member of an SNMP group. An SNMP *host* is the recipient of an SNMP trap operation. An SNMP *engine ID* is a name for the local or remote SNMP engine.

When configuring SNMP, follow these guidelines:

- When configuring an SNMP group, do not specify a notify view. The **snmp-server host** global configuration command auto-generates a notify view for the user and then adds it to the group associated with that user. Modifying the group's notify view affects all users associated with that group.
- To configure a remote user, specify the IP address or port number for the remote SNMP agent of the device where the user resides.
- Before you configure remote users for a particular agent, configure the SNMP engine ID, using the **snmp-server engineID** global configuration command with the **remote** option. The remote agent's

SNMP engine ID and user password are used to compute the authentication and privacy digests. If you do not configure the remote engine ID first, the configuration command fails.

- When configuring SNMP informs, you need to configure the SNMP engine ID for the remote agent in the SNMP database before you can send proxy requests or informs to it.
- If a local user is not associated with a remote host, the switch does not send informs for the **auth** (authNoPriv) and the **priv** (authPriv) authentication levels.
- Changing the value of the SNMP engine ID has significant results. A user's password (entered on the command line) is converted to an MD5 or SHA security digest based on the password and the local engine ID. The command-line password is then destroyed, as required by RFC 2274. Because of this deletion, if the value of the engine ID changes, the security digests of SNMPv3 users become invalid, and you need to reconfigure SNMP users by using the **snmp-server user** *username* global configuration command. Similar restrictions require the reconfiguration of community strings when the engine ID changes.

How to Configure SNMP

Disabling the SNMP Agent

The **no snmp-server** global configuration command disables all running versions (Version 1, Version 2C, and Version 3) of the SNMP agent on the device. You reenable all versions of the SNMP agent by the first **snmp-server** global configuration command that you enter. There is no Cisco IOS command specifically designated for enabling SNMP.

Beginning in privileged EXEC mode, follow these steps to disable the SNMP agent.

Before You Begin

The SNMP Agent must be enabled before it can be disabled. The SNMP agent is enabled by the first **snmp-server** global configuration command entered on the device.

SUMMARY STEPS

1. **configure terminal**
2. **no snmp-server**
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.

	Command or Action	Purpose
Step 2	**no snmp-server** **Example:** `Switch(config)# no snmp-server`	Disables the SNMP agent operation.
Step 3	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Configuring Community Strings

You use the SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to permit access to the agent on the switch. Optionally, you can specify one or more of these characteristics associated with the string:

- An access list of IP addresses of the SNMP managers that are permitted to use the community string to gain access to the agent
- A MIB view, which defines the subset of all MIB objects accessible to the given community
- Read and write or read-only permission for the MIB objects accessible to the community

Beginning in privileged EXEC mode, follow these steps to configure a community string on the switch.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*access-list-number*]
3. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.

	Command or Action	Purpose
Step 2	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*access-list-number*] **Example:** `Switch(config)# snmp-server community comaccess ro 4`	Configures the community string. **Note** The @ symbol is used for delimiting the context information. Avoid using the @ symbol as part of the SNMP community string when configuring this command. • For *string*, specify a string that acts like a password and permits access to the SNMP protocol. You can configure one or more community strings of any length. • (Optional) For *view-name*, specify the view record accessible to the community. • (Optional) Specify either read-only (**ro**) if you want authorized management stations to retrieve MIB objects, or specify read-write (**rw**) if you want authorized management stations to retrieve and modify MIB objects. By default, the community string permits read-only access to all objects. • (Optional) For *access-list-number*, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **Example:** `Switch(config)# access-list 4 deny any`	(Optional) If you specified an IP standard access list number in Step 2, then create the list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access if the conditions are matched. The **permit** keyword permits access if the conditions are matched. • For *source*, enter the IP address of the SNMP managers that are permitted to use the community string to gain access to the agent. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. Recall that the access list is always terminated by an implicit deny statement for everything.
Step 4	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

This example shows how to assign the comaccess string to SNMP, to allow read-only access, and to specify that IP access list 4 can use the community string to gain access to the switch SNMP agent:

```
Switch(config)# snmp-server community comaccess ro 4
```

What to Do Next

To disable access for an SNMP community, set the community string for that community to the null string (do not enter a value for the community string).

To remove a specific community string, use the **no snmp-server** community string global configuration command.

You can specify an identification name (engine ID) for the local or remote SNMP server engine on the switch. You can configure an SNMP server group that maps SNMP users to SNMP views, and you can add new users to the SNMP group.

Configuring SNMP Groups and Users

You can specify an identification name (engine ID) for the local or remote SNMP server engine on the switch. You can configure an SNMP server group that maps SNMP users to SNMP views, and you can add new users to the SNMP group.

Beginning in privileged EXEC mode, follow these steps to configure SNMP groups and users on the switch.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server engineID** {**local** *engineid-string* | **remote** *ip-address* [**udp-port** *port-number*] *engineid-string*}
3. **snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]
4. **snmp-server user** *username group-name* {**remote** *host* [**udp-port** *port*]} {**v1** [**access** *access-list*] | **v2c** [**access** *access-list*] | **v3** [**encrypted**] [**access** *access-list*] [**auth** {**md5** | **sha**} *auth-password*] } [*priv* {**des** | **3des** | **aes** {**128** | **192** | **256**}} *priv-password*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**snmp-server engineID** {**local** *engineid-string* | **remote** *ip-address* [**udp-port** *port-number*] *engineid-string*} **Example:** `Switch(config)# snmp-server engineID local 1234`	Configures a name for either the local or remote copy of SNMP. • The *engineid-string* is a 24-character ID string with the name of the copy of SNMP. You need not specify the entire 24-character engine ID if it has trailing zeros. Specify only the portion of the engine ID up to the point where only zeros remain in the value. The Step Example configures an engine ID of 123400000000000000000000.

	Command or Action	Purpose
		• If you select **remote**, specify the *ip-address* of the device that contains the remote copy of SNMP and the optional User Datagram Protocol (UDP) port on the remote device. The default is 162.
Step 3	**snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*] **Example:** `Switch(config)# snmp-server group public v2c access lmnop`	Configures a new SNMP group on the remote device. For *group-name*, specify the name of the group. Specify one of the following security models: • **v1** is the least secure of the possible security models. • **v2c** is the second least secure model. It allows transmission of informs and integers twice the normal width. • **v3**, the most secure, requires you to select one of the following authentication levels: **auth—**Enables the Message Digest 5 (MD5) and the Secure Hash Algorithm (SHA) packet authentication. **noauth—**Enables the noAuthNoPriv security level. This is the default if no keyword is specified. **priv—**Enables Data Encryption Standard (DES) packet encryption (also called privacy). (Optional) Enter **read** *readview* with a string (not to exceed 64 characters) that is the name of the view in which you can only view the contents of the agent. (Optional) Enter **write** *writeview* with a string (not to exceed 64 characters) that is the name of the view in which you enter data and configure the contents of the agent. (Optional) Enter **notify** *notifyview* with a string (not to exceed 64 characters) that is the name of the view in which you specify a notify, inform, or trap. (Optional) Enter **access** *access-list* with a string (not to exceed 64 characters) that is the name of the access list.
Step 4	**snmp-server user** *username group-name* {**remote** *host* [**udp-port** *port*]} {**v1** [**access** *access-list*] | **v2c** [**access** *access-list*] | **v3** [**encrypted**] [**access** *access-list*] [**auth** {**md5** | **sha**} *auth-password*] } [*priv* {**des** | **3des** | **aes** {**128** | **192** | **256**}} *priv-password*] **Example:** `Switch(config)# snmp-server user Pat public v2c`	Adds a new user for an SNMP group. The *username* is the name of the user on the host that connects to the agent. The *group-name* is the name of the group to which the user is associated. Enter **remote** to specify a remote SNMP entity to which the user belongs and the hostname or IP address of that entity with the optional UDP port number. The default is 162. Enter the SNMP version number (**v1**, **v2c**, or **v3**). If you enter **v3**, you have these additional options: • **encrypted** specifies that the password appears in encrypted format. This keyword is available only when the **v3** keyword is specified.

	Command or Action	Purpose
		• **auth** is an authentication level setting session that can be either the HMAC-MD5-96 (**md5**) or the HMAC-SHA-96 (**sha**) authentication level and requires a password string *auth-password* (not to exceed 64 characters). If you enter **v3** you can also configure a private (**priv**) encryption algorithm and password string *priv-password* using the following keywords (not to exceed 64 characters): • **priv** specifies the User-based Security Model (USM). • **des** specifies the use of the 56-bit DES algorithm. • **3des** specifies the use of the 168-bit DES algorithm. • **aes** specifies the use of the DES algorithm. You must select either 128-bit, 192-bit, or 256-bit encryption. (Optional) Enter **access** *access-list* with a string (not to exceed 64 characters) that is the name of the access list.
Step 5	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Configuring SNMP Notifications

A trap manager is a management station that receives and processes traps. Traps are system alerts that the switch generates when certain events occur. By default, no trap manager is defined, and no traps are sent. Switches running this Cisco IOS release can have an unlimited number of trap managers.

Note Many commands use the word **traps** in the command syntax. Unless there is an option in the command to select either traps or informs, the keyword **traps** refers to traps, informs, or both. Use the **snmp-server host** global configuration command to specify whether to send SNMP notifications as traps or informs.

You can use the **snmp-server host** global configuration command for a specific host to receive the notification types listed in the following table. You can enable any or all of these traps and configure a trap manager to receive them.

Table 10: Device Notification Types

Notification Type Keyword	Description
bridge	Generates STP bridge MIB traps.

Notification Type Keyword	Description
cluster	Generates a trap when the cluster configuration changes.
config	Generates a trap for SNMP configuration changes.
copy-config	Generates a trap for SNMP copy configuration changes.
cpu threshold	Allow CPU-related traps.
entity	Generates a trap for SNMP entity changes.
envmon	Generates environmental monitor traps. You can enable any or all of these environmental traps: fan, shutdown, status, supply, temperature.
flash	Generates SNMP FLASH notifications. In a switch stack, you can optionally enable notification for flash insertion or removal, which would cause a trap to be issued whenever a switch in the stack is removed or inserted (physical removal, power cycle, or reload).
fru-ctrl	Generates entity field-replaceable unit (FRU) control traps. In the switch stack, this trap refers to the insertion or removal of a switch in the stack.
hsrp	Generates a trap for Hot Standby Router Protocol (HSRP) changes.
ipmulticast	Generates a trap for IP multicast routing changes.
mac-notification	Generates a trap for MAC address notifications.
ospf	Generates a trap for Open Shortest Path First (OSPF) changes. You can enable any or all of these traps: Cisco specific, errors, link-state advertisement, rate limit, retransmit, and state changes.
pim	Generates a trap for Protocol-Independent Multicast (PIM) changes. You can enable any or all of these traps: invalid PIM messages, neighbor changes, and rendezvous point (RP)-mapping changes.
port-security	Generates SNMP port security traps. You can also set a maximum trap rate per second. The range is from 0 to 1000; the default is 0, which means that there is no rate limit. **Note** When you configure a trap by using the notification type **port-security**, configure the port security trap first, and then configure the port security trap rate: **1** **snmp-server enable traps port-security** **2** **snmp-server enable traps port-security trap-rate** *rate*
snmp	Generates a trap for SNMP-type notifications for authentication, cold start, warm start, link up or link down.
storm-control	Generates a trap for SNMP storm-control. You can also set a maximum trap rate per minute. The range is from 0 to 1000; the default is 0 (no limit is imposed; a trap is sent at every occurrence).

Notification Type Keyword	Description
stpx	Generates SNMP STP Extended MIB traps.
syslog	Generates SNMP syslog traps.
tty	Generates a trap for TCP connections. This trap is enabled by default.
vlan-membership	Generates a trap for SNMP VLAN membership changes.
vlancreate	Generates SNMP VLAN created traps.
vlandelete	Generates SNMP VLAN deleted traps.
vtp	Generates a trap for VLAN Trunking Protocol (VTP) changes.

Beginning in privileged EXEC mode, follow these steps to configure the switch to send traps or informs to a host.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server engineID remote** *ip-address engineid-string*
3. **snmp-server user** *username group-name* {**remote** *host* [**udp-port** *port*]} {**v1** [**access** *access-list*] | **v2c** [**access** *access-list*] | **v3** [**encrypted**] [**access** *access-list*] [**auth** {**md5** | **sha**} *auth-password*] }
4. **snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]
5. **snmp-server host** *host-addr* [**informs** | **traps**] [**version** {**1** | **2c** | **3** {**auth** | **noauth** | **priv**}}] *community-string* [*notification-type*]
6. **snmp-server enable traps** *notification-types*
7. **snmp-server trap-source** *interface-id*
8. **snmp-server queue-length** *length*
9. **snmp-server trap-timeout** *seconds*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**snmp-server engineID remote** *ip-address engineid-string*	Specifies the engine ID for the remote host.

	Command or Action	Purpose
	Example: `Switch(config)# snmp-server engineID remote 192.180.1.27 00000063000100a1c0b4011b`	
Step 3	**snmp-server user** *username group-name* {**remote** *host* [**udp-port** *port*]} {**v1** [**access** *access-list*] | **v2c** [**access** *access-list*] | **v3** [**encrypted**] [**access** *access-list*] [**auth** {**md5** | **sha**} *auth-password*] } **Example:** `Switch(config)# snmp-server user Pat public v2c`	Configures an SNMP user to be associated with the remote host created in Step 2. **Note** You cannot configure a remote user for an address without first configuring the engine ID for the remote host. Otherwise, you receive an error message, and the command is not executed.
Step 4	**snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*] **Example:** `Switch(config)# snmp-server group public v2c access lmnop`	Configures an SNMP group.
Step 5	**snmp-server host** *host-addr* [**informs** | **traps**] [**version** {**1** | **2c** | **3** {**auth** | **noauth** | **priv**}}] *community-string* [*notification-type*] **Example:** `Switch(config)# snmp-server host 203.0.113.1 comaccess snmp`	Specifies the recipient of an SNMP trap operation. For *host-addr*, specify the name or Internet address of the host (the targeted recipient). (Optional) Specify **traps** (the default) to send SNMP traps to the host. Specify **informs** to send SNMP informs to the host. (Optional) Specify the SNMP **version** (**1**, **2c**, or **3**). SNMPv1 does not support informs. (Optional) For Version 3, select authentication level **auth**, **noauth**, or **priv**. For *community-string*, when **version 1** or **version 2c** is specified, enter the password-like community string sent with the notification operation. When **version 3** is specified, enter the SNMPv3 username. The @ symbol is used for delimiting the context information. Avoid using the @ symbol as part of the SNMP community string when configuring this command. (Optional) For *notification-type*, use the keywords listed in the table above. If no type is specified, all notifications are sent.
Step 6	**snmp-server enable traps** *notification-types* **Example:** `Switch(config)# snmp-server enable traps snmp`	Enables the switch to send traps or informs and specifies the type of notifications to be sent. For a list of notification types, see the table above, or enter **snmp-server enable traps ?** To enable multiple types of traps, you must enter a separate **snmp-server enable traps** command for each trap type.

	Command or Action	Purpose
		Note When you configure a trap by using the notification type port-security, configure the port security trap first, and then configure the port security trap rate: **1 snmp-server enable traps port-security** **2 snmp-server enable traps port-security trap-rate** *rate*
Step 7	**snmp-server trap-source** *interface-id* **Example:** `Switch(config)# snmp-server trap-source GigabitEthernet1/0/1`	(Optional) Specifies the source interface, which provides the IP address for the trap message. This command also sets the source IP address for informs.
Step 8	**snmp-server queue-length** *length* **Example:** `Switch(config)# snmp-server queue-length 20`	(Optional) Establishes the message queue length for each trap host. The range is 1 to 1000; the default is 10.
Step 9	**snmp-server trap-timeout** *seconds* **Example:** `Switch(config)# snmp-server trap-timeout 60`	(Optional) Defines how often to resend trap messages. The range is 1 to 1000; the default is 30 seconds.
Step 10	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

What to Do Next

The **snmp-server host** command specifies which hosts receive the notifications. The **snmp-server enable trap** command globally enables the method for the specified notification (for traps and informs). To enable a host to receive an inform, you must configure an **snmp-server host informs** command for the host and globally enable informs by using the **snmp-server enable traps** command.

To remove the specified host from receiving traps, use the **no snmp-server host** *host* global configuration command. The **no snmp-server host** command with no keywords disables traps, but not informs, to the host. To disable informs, use the **no snmp-server host informs** global configuration command. To disable a specific trap type, use the **no snmp-server enable traps** *notification-types* global configuration command.

Setting the Agent Contact and Location Information

Beginning in privileged EXEC mode, follow these steps to set the system contact and location of the SNMP agent so that these descriptions can be accessed through the configuration file.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server contact** *text*
3. **snmp-server location** *text*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**snmp-server contact** *text* **Example:** `Switch(config)# snmp-server contact Dial System Operator at beeper 21555`	Sets the system contact string.
Step 3	**snmp-server location** *text* **Example:** `Switch(config)# snmp-server location Building 3/Room 222`	Sets the system location string.
Step 4	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Limiting TFTP Servers Used Through SNMP

Beginning in privileged EXEC mode, follow these steps to limit the TFTP servers used for saving and loading configuration files through SNMP to the servers specified in an access list.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server tftp-server-list** *access-list-number*
3. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**snmp-server tftp-server-list** *access-list-number* **Example:** `Switch(config)# snmp-server tftp-server-list 44`	Limits the TFTP servers used for configuration file copies through SNMP to the servers in the access list. For *access-list-number*, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **Example:** `Switch(config)# access-list 44 permit 10.1.1.2`	Creates a standard access list, repeating the command as many times as necessary. For *access-list-number*, enter the access list number specified in Step 2. The **deny** keyword denies access if the conditions are matched. The **permit** keyword permits access if the conditions are matched. For *source*, enter the IP address of the TFTP servers that can access the switch. (Optional) For *source-wildcard*, enter the wildcard bits, in dotted decimal notation, to be applied to the source. Place ones in the bit positions that you want to ignore. The access list is always terminated by an implicit deny statement for everything.
Step 4	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Configuring Trap Flags for SNMP

SUMMARY STEPS

1. **configure terminal**
2. **trapflags ap** { **interfaceup | register**}
3. **trapflags client** {**dot11 | excluded**}
4. **trapflags dot11-security** {**ids-sig-attack | wep-decrypt-error**}
5. **trapflags mesh**
6. **trapflags rogueap**
7. **trapflags rrm-params** {**channels | tx-power**}
8. **trapflags rrm-profile** {**coverage | interference | load | noise**}
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**trapflags ap** { **interfaceup	register**} **Example:** `Switch(config)# trapflags ap interfaceup`	Enables sending AP-related traps. Use the **no** form of the command to disable the trap flags. • **interfaceup–** Enables trap when a Cisco AP interface (A or B) comes up. • **register–** Enables trap when a Cisco AP registers with a Cisco switch.
Step 3	**trapflags client** {**dot11	excluded**} **Example:** `Switch(config)# trapflags client excluded`	Enables sending client-related dot11 traps. Use the **no** form of the command to disable the trap flags. • **dot11–** Enables Dot11 traps for clients. • **excluded–** Enables excluded traps for clients.
Step 4	**trapflags dot11-security** {**ids-sig-attack	wep-decrypt-error**} **Example:** `Switch(config)# trapflags dot11-security wep-decrypt-error`	Enables sending 802.11 security-related traps. Use the **no** form of the command to disable the trap flags. • **ids-sig-attack–** Enables IDS signature attack traps. • **wep-decrypt-error–** Enables traps for WEP decrypt error for clients.

	Command or Action	Purpose
Step 5	**trapflags mesh** **Example:** `Switch(config)# trapflags mesh`	Enables trap for the mesh. Use the **no** form of the command to disable the trap flags.
Step 6	**trapflags rogueap** **Example:** `Switch(config)# trapflags rogueap`	Enables trap for rogue AP detection. Use the **no** form of the command to disable the trap flags.
Step 7	**trapflags rrm-params** {**channels** | **tx-power**} **Example:** `Switch(config)# trapflags rrm-params tx-power`	Enables sending RRM-parameter update-related traps. Use the **no** form of the command to disable the trap flags. • **channels–** Enables trap when RF Manager automatically changes a channel number for the Cisco AP interface. • **tx-power–** Enables the trap when RF Manager automatically changes Tx-Power level for the Cisco AP interface.
Step 8	**trapflags rrm-profile** {**coverage** | **interference** | **load** | **noise**} **Example:** `Switch(config)# trapflags rrm-profile interference`	Enables sending RRM-profile-related traps. Use the **no** form of the command to disable the trap flags. • **coverage–** Enables the trap when the coverage profile maintained by RF Manager fails. • **interference–** Enables the trap when the interference profile maintained by RF Manager fails. • **load–** Enables trap when the load profile maintained by RF Manager fails. • **noise–** Enables trap when the noise profile maintained by RF Manager fails.
Step 9	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Enabling SNMP Wireless Trap Notification

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server enable traps wireless** [**AP** | **RRM** | **bsn80211SecurityTrap** | **bsnAPParamUpdate** | **bsnAPProfile** | **bsnAccessPoint** | **bsnMobileStation** | **bsnRogue** | **client** | **mfp** | **rogue**]
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**snmp-server enable traps wireless** [**AP** | **RRM** | **bsn80211SecurityTrap** | **bsnAPParamUpdate** | **bsnAPProfile** | **bsnAccessPoint** | **bsnMobileStation** | **bsnRogue** | **client** | **mfp** | **rogue**] **Example:** `Switch(config)# snmp-server enable traps wireless AP`	Enables SNMP wireless trap notification. • **AP–** Enables access point traps. • **RRM–** Enables RRM traps. • **bsn80211SecurityTrap–** Enables the security-related trap. • **bsnAPParamUpdate–** Enables the trap for AP parameters that get updated. • **bsnAPProfile–** Enables BSN AP profile traps. • **bsnAccessPoint–** Enables BSN access point traps. • **bsnMobileStation–** Controls wireless client traps. • **bsnRogue–** Enables BSN rogue-related traps. • **client–** Enables client traps. • **mfp–** Enables MFP traps. • **rogue–** Enables rogue-related traps.
Step 3	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Monitoring SNMP Status

To display SNMP input and output statistics, including the number of illegal community string entries, errors, and requested variables, use the **show snmp** privileged EXEC command. You also can use the other privileged EXEC commands listed in the table to display SNMP information.

Table 11: Commands for Displaying SNMP Information

Command	Purpose
show snmp	Displays SNMP statistics.
show snmp engineID	Displays information on the local SNMP engine and all remote engines that have been configured on the device.
show snmp group	Displays information on each SNMP group on the network.
show snmp pending	Displays information on pending SNMP requests.
show snmp sessions	Displays information on the current SNMP sessions.
show snmp user	Displays information on each SNMP user name in the SNMP users table. **Note** You must use this command to display SNMPv3 configuration information for **auth** | **noauth** | **priv** mode. This information is not displayed in the **show running-config** output.

SNMP Examples

This example shows how to enable all versions of SNMP. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string *public*. This configuration does not cause the switch to send any traps.

```
Switch(config)# snmp-server community public
```

This example shows how to permit any SNMP manager to access all objects with read-only permission using the community string *public*. The switch also sends VTP traps to the hosts 192.180.1.111 and 192.180.1.33 using SNMPv1 and to the host 192.180.1.27 using SNMPv2C. The community string *public* is sent with the traps.

```
Switch(config)# snmp-server community public
Switch(config)# snmp-server enable traps vtp
Switch(config)# snmp-server host 192.180.1.27 version 2c public
Switch(config)# snmp-server host 192.180.1.111 version 1 public
Switch(config)# snmp-server host 192.180.1.33 public
```

This example shows how to allow read-only access for all objects to members of access list 4 that use the *comaccess* community string. No other SNMP managers have access to any objects. SNMP Authentication Failure traps are sent by SNMPv2C to the host *cisco.com* using the community string *public*.

```
Switch(config)# snmp-server community comaccess ro 4
Switch(config)# snmp-server enable traps snmp authentication
Switch(config)# snmp-server host cisco.com version 2c public
```

This example shows how to send Entity MIB traps to the host *cisco.com*. The community string is restricted. The first line enables the switch to send Entity MIB traps in addition to any traps previously enabled. The second line specifies the destination of these traps and overwrites any previous **snmp-server** host commands for the host *cisco.com*.

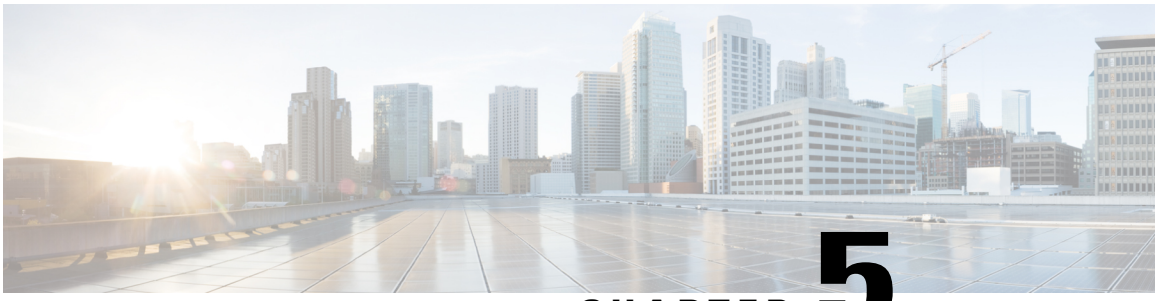
```
Switch(config)# snmp-server enable traps entity
Switch(config)# snmp-server host cisco.com restricted entity
```

This example shows how to enable the switch to send all traps to the host *myhost.cisco.com* using the community string *public*:

```
Switch(config)# snmp-server enable traps
Switch(config)# snmp-server host myhost.cisco.com public
```

This example shows how to associate a user with a remote host and to send **auth** (authNoPriv) authentication-level informs when the user enters global configuration mode:

```
Switch(config)# snmp-server engineID remote 192.180.1.27 00000063000100a1c0b4011b
Switch(config)# snmp-server group authgroup v3 auth
Switch(config)# snmp-server user authuser authgroup remote 192.180.1.27 v3 auth md5 mypassword
Switch(config)# snmp-server user authuser authgroup v3 auth md5 mypassword
Switch(config)# snmp-server host 192.180.1.27 informs version 3 auth authuser config
Switch(config)# snmp-server enable traps
Switch(config)# snmp-server inform retries 0
```



CHAPTER 5

Configuring Service Level Agreements

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions on SLAs

This section lists the restrictions on SLAs.

The following are restrictions on IP SLAs network performance measurement:

- The switch does not support VoIP service levels using the gatekeeper registration delay operations measurements.
- Only a Cisco IOS device can be a source for a destination IP SLAs responder.
- You cannot configure the IP SLAs responder on non-Cisco devices and Cisco IOS IP SLAs can send operational packets only to services native to those devices.

Related Topics

Implementing IP SLA Network Performance Measurement, on page 77
Network Performance Measurement with Cisco IOS IP SLAs, on page 71
IP SLA Responder and IP SLA Control Protocol, on page 72

Information About SLAs

Cisco IOS IP Service Level Agreements (SLAs)

Cisco IOS IP SLAs send data across the network to measure performance between multiple network locations or across multiple network paths. They simulate network data and IP services and collect network performance information in real time. Cisco IOS IP SLAs generate and analyze traffic either between Cisco IOS devices or from a Cisco IOS device to a remote IP device such as a network application server. Measurements provided by the various Cisco IOS IP SLA operations can be used for troubleshooting, for problem analysis, and for designing network topologies.

Depending on the specific Cisco IOS IP SLA operations, various network performance statistics are monitored within the Cisco device and stored in both command-line interface (CLI) and Simple Network Management Protocol (SNMP) MIBs. IP SLA packets have configurable IP and application layer options such as source and destination IP address, User Datagram Protocol (UDP)/TCP port numbers, a type of service (ToS) byte (including Differentiated Services Code Point [DSCP] and IP Prefix bits), Virtual Private Network (VPN) routing/forwarding instance (VRF), and URL web address.

Because Cisco IP SLAs are Layer 2 transport independent, you can configure end-to-end operations over disparate networks to best reflect the metrics that an end user is likely to experience. IP SLAs collect a unique subset of the following performance metrics:

- Delay (both round-trip and one-way)
- Jitter (directional)
- Packet loss (directional)
- Packet sequencing (packet ordering)
- Path (per hop)
- Connectivity (directional)
- Server or website download time

Because Cisco IOS IP SLAs is SNMP-accessible, it can also be used by performance-monitoring applications like Cisco Prime Internetwork Performance Monitor (IPM) and other third-party Cisco partner performance management products.

Using IP SLAs can provide the following benefits:

- Service-level agreement monitoring, measurement, and verification.
- Network performance monitoring
    - Measurement of jitter, latency, or packet loss in the network.
    - Continuous, reliable, and predictable measurements.

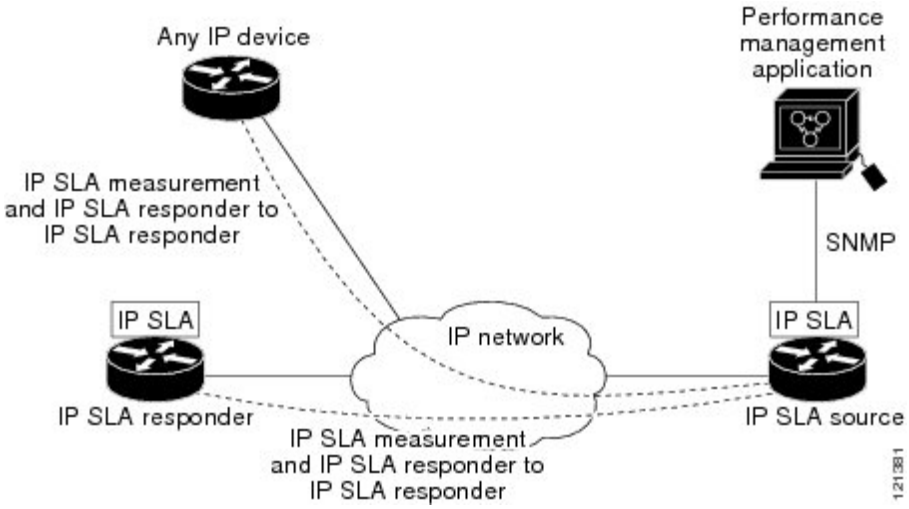
- IP service network health assessment to verify that the existing QoS is sufficient for new IP services.
- Edge-to-edge network availability monitoring for proactive verification and connectivity testing of network resources (for example, shows the network availability of an NFS server used to store business critical data from a remote site).
- Network operation troubleshooting by providing consistent, reliable measurement that immediately identifies problems and saves troubleshooting time.
- Multiprotocol Label Switching (MPLS) performance monitoring and network verification (if the switch supports MPLS).

Network Performance Measurement with Cisco IOS IP SLAs

You can use IP SLAs to monitor the performance between any area in the network—core, distribution, and edge—without deploying a physical probe. It uses generated traffic to measure network performance between two networking devices.

The following figure shows how IP SLAs begin when the source device sends a generated packet to the destination device. After the destination device receives the packet, depending on the type of IP SLAs operation, it responds with time-stamp information for the source to make the calculation on performance metrics. An IP SLAs operation performs a network measurement from the source device to a destination in the network using a specific protocol such as UDP.

Figure 4: Cisco IOS IP SLAs Operation



Related Topics

Implementing IP SLA Network Performance Measurement, on page 77

Restrictions on SLAs, on page 69

IP SLA Responder and IP SLA Control Protocol

The IP SLA responder is a component embedded in the destination Cisco device that allows the system to anticipate and respond to IP SLA request packets. The responder provides accurate measurements without the need for dedicated probes. The responder uses the Cisco IOS IP SLA Control Protocol to provide a mechanism through which it can be notified on which port it should listen and respond.

Note The IP SLA responder can be a Cisco IOS Layer 2, responder-configurable switch. The responder does not need to support full IP SLA functionality.

The following figure shows where the Cisco IOS IP SLA responder fits in the IP network. The responder listens on a specific port for control protocol messages sent by an IP SLA operation. Upon receipt of the control message, it enables the specified UDP or TCP port for the specified duration. During this time, the responder accepts the requests and responds to them. It disables the port after it responds to the IP SLA packet, or when the specified time expires. MD5 authentication for control messages is available for added security.

You do not need to enable the responder on the destination device for all IP SLA operations. For example, a responder is not required for services that are already provided by the destination router (such as Telnet or HTTP).

Related Topics

Restrictions on SLAs, on page 69

Response Time Computation for IP SLAs

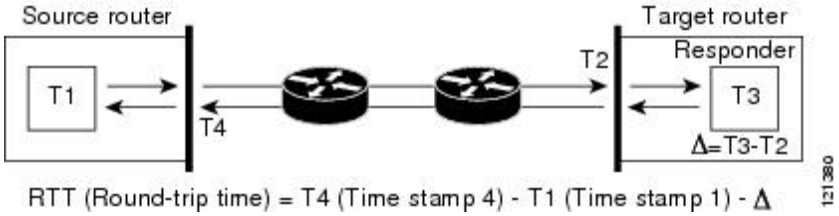
Switches, controllers, and routers can take tens of milliseconds to process incoming packets due to other high priority processes. This delay affects the response times because the test-packet reply might be in a queue while waiting to be processed. In this situation, the response times would not accurately represent true network delays. IP SLAs minimize these processing delays on the source device as well as on the target device (if the responder is being used) to determine true round-trip times. IP SLA test packets use time stamping to minimize the processing delays.

When the IP SLA responder is enabled, it allows the target device to take time stamps when the packet arrives on the interface at interrupt level and again just as it is leaving, eliminating the processing time. This time stamping is made with a granularity of sub-milliseconds (ms).

The following figure demonstrates how the responder works. Four time stamps are taken to make the calculation for round-trip time. At the target router, with the responder functionality enabled, time stamp 2 (TS2) is subtracted from time stamp 3 (TS3) to produce the time spent processing the test packet as represented by delta. This delta value is then subtracted from the overall round-trip time. Notice that the same principle is

applied by IP SLAs on the source router where the incoming time stamp 4 (TS4) is also taken at the interrupt level to allow for greater accuracy.

Figure 5: Cisco IOS IP SLA Responder Time Stamping



An additional benefit of the two time stamps at the target device is the ability to track one-way delay, jitter, and directional packet loss. Because much network behavior is asynchronous, it is critical to have these statistics. However, to capture one-way delay measurements, you must configure both the source router and target router with Network Time Protocol (NTP) so that the source and target are synchronized to the same clock source. One-way jitter measurements do not require clock synchronization.

IP SLAs Operation Scheduling

When you configure an IP SLAs operation, you must schedule the operation to begin capturing statistics and collecting error information. You can schedule an operation to start immediately or to start at a certain month, day, and hour. You can use the *pending* option to set the operation to start at a later time. The pending option is an internal state of the operation that is visible through SNMP. The pending state is also used when an operation is a reaction (threshold) operation waiting to be triggered. You can schedule a single IP SLAs operation or a group of operations at one time.

You can schedule several IP SLAs operations by using a single command through the Cisco IOS CLI or the CISCO RTTMON-MIB. Scheduling the operations to run at evenly distributed times allows you to control the amount of IP SLAs monitoring traffic. This distribution of IP SLA operations helps minimize the CPU utilization and thus improves network scalability.

For more details about the IP SLA multi-operations scheduling functionality, see the "IP SLAs—Multiple Operation Scheduling" chapter of the *Cisco IOS IP SLAs Configuration Guide*.

IP SLA Operation Threshold Monitoring

To support successful service level agreement monitoring, you must have mechanisms that notify you immediately of any possible violation. IP SLAs can send SNMP traps that are triggered by events such as the following:

- Connection loss
- Timeout
- Round-trip time threshold
- Average jitter threshold
- One-way packet loss
- One-way jitter

- One-way mean opinion score (MOS)
- One-way latency

An IP SLA threshold violation can also trigger another IP SLA operation for further analysis. For example, the frequency could be increased or an Internet Control Message Protocol (ICMP) path echo or ICMP path jitter operation could be initiated for troubleshooting.

ICMP Echo

The ICMP echo operation measures the end-to-end response time between a Cisco device and any other device that uses IP. The response time is computed by measuring the time it takes to send an ICMP echo request message to a destination and receive an ICMP echo reply. Many customers use IP SLA ICMP-based operations, in-house ping testing, or ping-based dedicated probes to measure this response time. The IP SLA ICMP echo operation conforms to the same specifications as ICMP ping testing, and both methods result in the same response times.

Related Topics

Analyzing IP Service Levels by Using the ICMP Echo Operation, on page 83

UDP Jitter

Jitter is a simple term that describes interpacket delay variance. When multiple packets are sent consecutively at an interval of 10 ms from source to destination, the destination should receive them 10 ms apart (if the network is behaving correctly). However, if there are delays in the network (such as queuing, arriving through alternate routes, and so on), the time interval between packet arrivals might be more or less than 10 ms. A positive jitter value indicates that the packets arrived more than 10 ms apart. A negative jitter value indicates that the packets arrived less than 10 ms apart. If the packets arrive 12 ms apart, the positive jitter is 2 ms; if the packets arrive 8 ms apart, the negative jitter is 2 ms. For delay-sensitive networks, positive jitter values are undesirable, and a jitter value of 0 is ideal.

In addition to monitoring jitter, the IP SLA UDP jitter operation can be used as a multipurpose data gathering operation. The packets generated by IP SLAs carry sequence information and time stamps from the source and operational target that include packet sending and receiving data. Based on this data, UDP jitter operations measure the following:

- Per-direction jitter (source to destination and destination to source)
- Per-direction packet-loss
- Per-direction delay (one-way delay)
- Round-trip delay (average round-trip time)

Because the paths for the sending and receiving of data can be different (asymmetric), you can use the per-direction data to more readily identify where congestion or other problems are occurring in the network.

The UDP jitter operation generates synthetic (simulated) UDP traffic and sends a number of UDP packets, each of a specified size, sent a specified number of milliseconds apart, from a source router to a target router, at a given frequency. By default, ten packet-frames, each with a payload size of 10 bytes are generated every 10 ms, and the operation is repeated every 60 seconds. You can configure each of these parameters to best simulate the IP service you want to provide.

To provide accurate one-way delay (latency) measurements, time synchronization (as provided by NTP) is required between the source and the target device. Time synchronization is not required for the one-way jitter and packet loss measurements. If the time is not synchronized between the source and target devices, one-way jitter and packet loss data is returned, but values of 0 are returned for the one-way delay measurements provided by the UDP jitter operation.

Related Topics

Analyzing IP Service Levels by Using the UDP Jitter Operation, on page 80

Configuration Guidelines

For information on the IP SLA commands, see the *Cisco IOS IP SLAs Command Reference, Release 12.4T* command reference.

For detailed descriptions and configuration procedures, see the *Cisco IOS IP SLAs Configuration Guide, Release 12.4TL*.

Not all of the IP SLA commands or operations described in the referenced guide are supported on the switch. The switch supports IP service level analysis by using UDP jitter, UDP echo, HTTP, TCP connect, ICMP echo, ICMP path echo, ICMP path jitter, FTP, DNS, and DHCP, as well as multiple operation scheduling and proactive threshold monitoring. It does not support VoIP service levels using the gatekeeper registration delay operations measurements.

Before configuring any IP SLAs application, you can use the **show ip sla application** privileged EXEC command to verify that the operation type is supported on your software image. This is an example of the output from the command:

```
Switch# show ip sla application

        IP Service Level Agreements
Version: Round Trip Time MIB 2.2.0, Infrastructure Engine-III

Supported Operation Types:
        icmpEcho, path-echo, path-jitter, udpEcho, tcpConnect, http
        dns, udpJitter, dhcp, ftp, udpApp, wspApp

Supported Features:
        IPSLAs Event Publisher

IP SLAs low memory water mark: 33299323
Estimated system max number of entries: 24389

Estimated number of configurable operations: 24389
Number of Entries configured  : 0
Number of active Entries      : 0
Number of pending Entries     : 0
Number of inactive Entries    : 0
Time of last change in whole IP SLAs: *13:04:37.668 UTC Wed Dec 19 2012
```

How to Configure IP SLAs Operations

This section does not include configuration information for all available operations as the configuration information details are included in the *Cisco IOS IP SLAs Configuration Guide*. It does include several operations as examples, including configuring the responder, configuring a UDP jitter operation, which requires

a responder, and configuring an ICMP echo operation, which does not require a responder. For details about configuring other operations, see the *Cisco IOS IP SLAs Configuration Guide*.

Configuring the IP SLA Responder

The IP SLA responder is available only on Cisco IOS software-based devices, including some Layer 2 switches that do not support full IP SLA functionality.

Beginning in privileged EXEC mode, follow these steps to configure the IP SLA responder on the target device (the operational target):

SUMMARY STEPS

1. **configure terminal**
2. **ip sla responder** {**tcp-connect** | **udp-echo**} **ipaddress** *ip-address* **port** *port-number*
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**ip sla responder** {**tcp-connect** | **udp-echo**} **ipaddress** *ip-address* **port** *port-number* **Example:** `Switch(config)# ip sla responder udp-echo 172.29.139.134 5000`	Configures the switch as an IP SLA responder. The keywords have these meanings: • **tcp-connect**—Enables the responder for TCP connect operations. • **udp-echo**—Enables the responder for User Datagram Protocol (UDP) echo or jitter operations. • **ipaddress** *ip-address*—Enter the destination IP address. • **port** *port-number*—Enter the destination port number. **Note** The IP address and port number must match those configured on the source device for the IP SLA operation.
Step 3	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

UDP Jitter Example

This example shows how to configure the device as a responder for the UDP jitter IP SLA operation in the next procedure:

```
Switch(config)# ip sla responder udp-echo 172.29.139.134 5000
```

Implementing IP SLA Network Performance Measurement

Beginning in privileged EXEC mode, follow these steps to implement IP SLA network performance measurement on your switch:

Before You Begin

Use the **show ip sla application** privileged EXEC command to verify that the desired operation type is supported on your software image.

SUMMARY STEPS

1. **configure terminal**
2. **ip sla** *operation-number*
3. **udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*]
4. **frequency** *seconds*
5. **threshold** *milliseconds*
6. **exit**
7. **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**ip sla** *operation-number* **Example:** `Switch(config)# ip sla 10`	Creates an IP SLA operation, and enters IP SLA configuration mode.

	Command or Action	Purpose
Step 3	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*] **Example:** `Switch(config-ip-sla)# udp-jitter 172.29.139.134 5000`	Configures the IP SLA operation as the operation type of your choice (a UDP jitter operation is used in the example), and enters its configuration mode (UDP jitter configuration mode is used in the example). • *destination-ip-address* | *destination-hostname*—Specifies the destination IP address or hostname. • *destination-port*—Specifies the destination port number in the range from 1 to 65535. • (Optional) **source-ip** {*ip-address* | *hostname*}—Specifies the source IP address or hostname. When a source IP address or hostname is not specified, IP SLA chooses the IP address nearest to the destination • (Optional) **source-port** *port-number*—Specifies the source port number in the range from 1 to 65535. When a port number is not specified, IP SLA chooses an available port. • (Optional) **control**—Enables or disables sending of IP SLA control messages to the IP SLA responder. By default, IP SLA control messages are sent to the destination device to establish a connection with the IP SLA responder • (Optional) **num-packets** *number-of-packets*—Enters the number of packets to be generated. The range is 1 to 6000; the default is 10. • (Optional) **interval** *inter-packet-interval*—Enters the interval between sending packets in milliseconds. The range is 1 to 6000; the default value is 20 ms.
Step 4	**frequency** *seconds* **Example:** `Switch(config-ip-sla-jitter)# frequency 45`	(Optional) Configures options for the SLA operation. This example sets the rate at which a specified IP SLA operation repeats. The range is from 1 to 604800 seconds; the default is 60 seconds.
Step 5	**threshold** *milliseconds* **Example:** `Switch(config-ip-sla-jitter)# threshold 200`	(Optional) Configures threshold conditions. This example sets the threshold of the specified IP SLA operation to 200. The range is from 0 to 60000 milliseconds.
Step 6	**exit** **Example:** `Switch(config-ip-sla-jitter)# exit`	Exits the SLA operation configuration mode (UDP jitter configuration mode in this example), and returns to global configuration mode.
Step 7	**ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm*	Configures the scheduling parameters for an individual IP SLA operation.

	Command or Action	Purpose
	[*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**] **Example:** `Switch(config)# ip sla schedule 10 start-time now life forever`	• *operation-number*—Enter the RTR entry number. • (Optional) **life**—Sets the operation to run indefinitely (**forever**) or for a specific number of *seconds*. The range is from 0 to 2147483647. The default is 3600 seconds (1 hour). • (Optional) **start-time**—Enters the time for the operation to begin collecting information: To start at a specific time, enter the hour, minute, second (in 24-hour notation), and day of the month. If no month is entered, the default is the current month. Enter **pending** to select no information collection until a start time is selected. Enter **now** to start the operation immediately. Enter **after** *hh:mm:ss* to show that the operation should start after the entered time has elapsed. • (Optional) **ageout** *seconds*—Enter the number of seconds to keep the operation in memory when it is not actively collecting information. The range is 0 to 2073600 seconds, the default is 0 seconds (never ages out). • (Optional) **recurring**—Set the operation to automatically run every day.
Step 8	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

UDP Jitter Configuration

This example shows how to configure a UDP jitter IP SLA operation:

```
Switch(config)# ip sla 10
Switch(config-ip-sla)# udp-jitter 172.29.139.134 5000
Switch(config-ip-sla-jitter)# frequency 30
Switch(config-ip-sla-jitter)# exit
Switch(config)# ip sla schedule 5 start-time now life forever
Switch(config)# end
Switch# show ip sla configuration 10
IP SLAs, Infrastructure Engine-II.

Entry number: 10
Owner:
Tag:
Type of operation to perform: udp-jitter
Target address/Source address: 1.1.1.1/0.0.0.0
Target port/Source port: 2/0
Request size (ARR data portion): 32
Operation timeout (milliseconds): 5000
Packet Interval (milliseconds)/Number of packets: 20/10
```

```
Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
    Operation frequency (seconds): 30
    Next Scheduled Start Time: Pending trigger
    Group Scheduled : FALSE
    Randomly Scheduled : FALSE
    Life (seconds): 3600
    Entry Ageout (seconds): never
    Recurring (Starting Everyday): FALSE
    Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 5000
Distribution Statistics:
    Number of statistic hours kept: 2
    Number of statistic distribution buckets kept: 1
    Statistic distribution interval (milliseconds): 20
Enhanced History:
```

Related Topics

Network Performance Measurement with Cisco IOS IP SLAs, on page 71

Restrictions on SLAs, on page 69

Analyzing IP Service Levels by Using the UDP Jitter Operation

Beginning in privileged EXEC mode, follow these steps to configure a UDP jitter operation on the source device:

Before You Begin

You must enable the IP SLA responder on the target device (the operational target) to configure a UDP jitter operation on the source device.

SUMMARY STEPS

1. **configure terminal**
2. **ip sla** *operation-number*
3. **udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*]
4. **frequency** *seconds*
5. **exit**
6. **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**ip sla** *operation-number* **Example:** `Switch(config)# ip sla 10`	Creates an IP SLA operation, and enters IP SLA configuration mode.
Step 3	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*] **Example:** `Switch(config-ip-sla)# udp-jitter 172.29.139.134 5000`	Configures the IP SLA operation as a UDP jitter operation, and enters UDP jitter configuration mode. • *destination-ip-address* | *destination-hostname*—Specifies the destination IP address or hostname. • *destination-port*—Specifies the destination port number in the range from 1 to 65535. • (Optional) **source-ip** {*ip-address* | *hostname*}—Specifies the source IP address or hostname. When a source IP address or hostname is not specified, IP SLA chooses the IP address nearest to the destination. • (Optional) **source-port** *port-number*—Specifies the source port number in the range from 1 to 65535. When a port number is not specified, IP SLA chooses an available port. • (Optional) **control**—Enables or disables sending of IP SLA control messages to the IP SLA responder. By default, IP SLA control messages are sent to the destination device to establish a connection with the IP SLA responder. • (Optional) **num-packets** *number-of-packets*—Enters the number of packets to be generated. The range is 1 to 6000; the default is 10. • (Optional) **interval** *inter-packet-interval*—Enters the interval between sending packets in milliseconds. The range is 1 to 6000; the default value is 20 ms.
Step 4	**frequency** *seconds* **Example:** `Switch(config-ip-sla-jitter)# frequency 45`	(Optional) Sets the rate at which a specified IP SLA operation repeats. The range is from 1 to 604800 seconds; the default is 60 seconds.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Switch(config-ip-sla-jitter)# exit`	Exits UDP jitter configuration mode, and returns to global configuration mode.
Step 6	**ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**] **Example:** `Switch(config)# ip sla schedule 10 start-time now life forever`	Configures the scheduling parameters for an individual IP SLA operation. • *operation-number*—Enter the RTR entry number. • (Optional) **life**—Sets the operation to run indefinitely (**forever**) or for a specific number of *seconds*. The range is from 0 to 2147483647. The default is 3600 seconds (1 hour). • (Optional) **start-time**—Enters the time for the operation to begin collecting information: To start at a specific time, enter the hour, minute, second (in 24-hour notation), and day of the month. If no month is entered, the default is the current month. Enter **pending** to select no information collection until a start time is selected. Enter **now** to start the operation immediately. Enter **after** *hh:mm:ss* to show that the operation should start after the entered time has elapsed. • (Optional) **ageout** *seconds*—Enter the number of seconds to keep the operation in memory when it is not actively collecting information. The range is 0 to 2073600 seconds, the default is 0 seconds (never ages out). • (Optional) **recurring**—Set the operation to automatically run every day.
Step 7	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Configuring a UDP Jitter IP SLA Operation

This example shows how to configure a UDP jitter IP SLA operation:

```
Switch(config)# ip sla 10
Switch(config-ip-sla)# udp-jitter 172.29.139.134 5000
Switch(config-ip-sla-jitter)# frequency 30
Switch(config-ip-sla-jitter)# exit
Switch(config)# ip sla schedule 5 start-time now life forever
Switch(config)# end
Switch# show ip sla configuration 10
```

```
IP SLAs, Infrastructure Engine-II.

Entry number: 10
Owner:
Tag:
Type of operation to perform: udp-jitter
Target address/Source address: 1.1.1.1/0.0.0.0
Target port/Source port: 2/0
Request size (ARR data portion): 32
Operation timeout (milliseconds): 5000
Packet Interval (milliseconds)/Number of packets: 20/10
Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
    Operation frequency (seconds): 30
    Next Scheduled Start Time: Pending trigger
    Group Scheduled : FALSE
    Randomly Scheduled : FALSE
    Life (seconds): 3600
    Entry Ageout (seconds): never
    Recurring (Starting Everyday): FALSE
    Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 5000
Distribution Statistics:
    Number of statistic hours kept: 2
    Number of statistic distribution buckets kept: 1
    Statistic distribution interval (milliseconds): 20
Enhanced History:
```

Related Topics

UDP Jitter, on page 74

Analyzing IP Service Levels by Using the ICMP Echo Operation

Beginning in privileged EXEC mode, follow these steps to configure an ICMP echo operation on the source device:

Before You Begin

This operation does not require the IP SLA responder to be enabled.

SUMMARY STEPS

1. **configure terminal**
2. **ip sla** *operation-number*
3. **icmp-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*} | **source-interface** *interface-id*]
4. **frequency** *seconds*
5. **exit**
6. **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**ip sla** *operation-number* **Example:** `Switch(config)# ip sla 10`	Creates an IP SLA operation and enters IP SLA configuration mode.
Step 3	**icmp-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*} | **source-interface** *interface-id*] **Example:** `Switch(config-ip-sla)# icmp-echo 172.29.139.134`	Configures the IP SLA operation as an ICMP Echo operation and enters ICMP echo configuration mode. • *destination-ip-address* | *destination-hostname*—Specifies the destination IP address or hostname. • (Optional) **source-ip** {*ip-address* | *hostname*}—Specifies the source IP address or hostname. When a source IP address or hostname is not specified, IP SLA chooses the IP address nearest to the destination. • (Optional) **source-interface** *interface-id*—Specifies the source interface for the operation.
Step 4	**frequency** *seconds* **Example:** `Switch(config-ip-sla-echo)# frequency 30`	(Optional) Sets the rate at which a specified IP SLA operation repeats. The range is from 1 to 604800 seconds; the default is 60 seconds.
Step 5	**exit** **Example:** `Switch(config-ip-sla-echo)# exit`	Exits UDP echo configuration mode, and returns to global configuration mode.
Step 6	**ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**] **Example:** `Switch(config)# ip sla schedule 5 start-time now life forever`	Configures the scheduling parameters for an individual IP SLA operation. • *operation-number*—Enter the RTR entry number. • (Optional) **life**—Sets the operation to run indefinitely (**forever**) or for a specific number of *seconds*. The range is from 0 to 2147483647. The default is 3600 seconds (1 hour) • (Optional) **start-time**—Enter the time for the operation to begin collecting information:

	Command or Action	Purpose
		To start at a specific time, enter the hour, minute, second (in 24-hour notation), and day of the month. If no month is entered, the default is the current month. Enter **pending** to select no information collection until a start time is selected. Enter **now** to start the operation immediately. Enter **after** *hh:mm:ss* to indicate that the operation should start after the entered time has elapsed. • (Optional) **ageout** *seconds*—Enter the number of seconds to keep the operation in memory when it is not actively collecting information. The range is 0 to 2073600 seconds; the default is 0 seconds (never ages out). • (Optional) **recurring**—Sets the operation to automatically run every day.
Step 7	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Configuring an ICMP Echo IP SLA Operation

This example shows how to configure an ICMP echo IP SLA operation:

```
Switch(config)# ip sla 12
Switch(config-ip-sla)# icmp-echo 172.29.139.134
Switch(config-ip-sla-echo)# frequency 30
Switch(config-ip-sla-echo)# exit
Switch(config)# ip sla schedule 5 start-time now life forever
Switch(config)# end
Switch# show ip sla configuration 22
IP SLAs, Infrastructure Engine-II.

Entry number: 12
Owner:
Tag:
Type of operation to perform: echo
Target address: 2.2.2.2
Source address: 0.0.0.0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Schedule:
    Operation frequency (seconds): 60
    Next Scheduled Start Time: Pending trigger
    Group Scheduled : FALSE
    Randomly Scheduled : FALSE
    Life (seconds): 3600
```

```
    Entry Ageout (seconds): never
    Recurring (Starting Everyday): FALSE
    Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 5000
Distribution Statistics:
    Number of statistic hours kept: 2
    Number of statistic distribution buckets kept: 1
    Statistic distribution interval (milliseconds): 20
History Statistics:
    Number of history Lives kept: 0
    Number of history Buckets kept: 15
    History Filter Type: None
Enhanced History:
```

Related Topics

IP SLA Operation Threshold Monitoring, on page 73

Monitoring IP SLA Operations

The following table describes the commands used to display IP SLA operation configurations and results:

Table 12: Monitoring IP SLA Operations

show ip sla application	Displays global information about Cisco IOS IP SLAs.
show ip sla authentication	Displays IP SLA authentication information.
show ip sla configuration [*entry-number*]	Displays configuration values including all defaults for all IP SLA operations or a specific operation.
show ip sla enhanced-history {**collection-statistics** | **distribution statistics**} [*entry-number*]	Displays enhanced history statistics for collected history buckets or distribution statistics for all IP SLA operations or a specific operation.
show ip sla ethernet-monitor configuration [*entry-number*]	Displays IP SLA automatic Ethernet configuration.
show ip sla group schedule [*schedule-entry-number*]	Displays IP SLA group scheduling configuration and details.
show ip sla history [*entry-number* | **full** | **tabular**]	Displays history collected for all IP SLA operations.
show ip sla mpls-lsp-monitor {**collection-statistics** | **configuration** | **ldp operational-state** | **scan-queue** | **summary** [*entry-number*] | **neighbors**}	Displays MPLS label switched path (LSP) Health Monitor operations.

show ip sla reaction-configuration [*entry-number*]	Displays the configured proactive threshold monitoring settings for all IP SLA operations or a specific operation.
show ip sla reaction-trigger [*entry-number*]	Displays the reaction trigger information for all IP SLA operations or a specific operation.
show ip sla responder	Displays information about the IP SLA responder.
show ip sla statistics [*entry-number* | **aggregated** | **details**]	Displays current or aggregated operational status and statistics.

Monitoring IP SLA Operation Examples

The following example shows all IP SLAs by application:

```
Switch# show ip sla application

        IP Service Level Agreements
Version: Round Trip Time MIB 2.2.0, Infrastructure Engine-III

Supported Operation Types:
        icmpEcho, path-echo, path-jitter, udpEcho, tcpConnect, http
        dns, udpJitter, dhcp, ftp, udpApp, wspApp

Supported Features:
        IPSLAs Event Publisher

IP SLAs low memory water mark: 33299323
Estimated system max number of entries: 24389

Estimated number of configurable operations: 24389
Number of Entries configured  : 0
Number of active Entries      : 0
Number of pending Entries     : 0
Number of inactive Entries    : 0
Time of last change in whole IP SLAs: *13:04:37.668 UTC Wed Dec 19 2012
```

The following example shows all IP SLA distribution statistics:

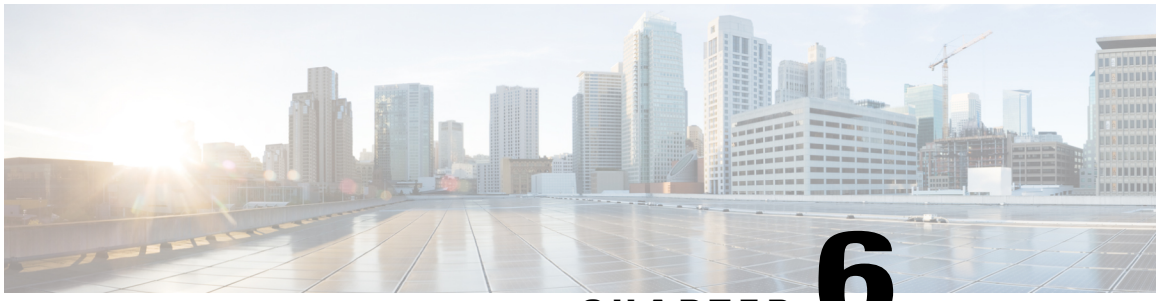
```
Switch# show ip sla enhanced-history distribution-statistics

Point by point Enhanced History
Entry    = Entry Number
Int      = Aggregation Interval
BucI     = Bucket Index
StartT   = Aggregation Start Time
Pth      = Path index
Hop      = Hop in path index
Comps    = Operations completed
OvrTh    = Operations completed over thresholds
SumCmp   = Sum of RTT (milliseconds)
SumCmp2L = Sum of RTT squared low 32 bits (milliseconds)
SumCmp2H = Sum of RTT squared high 32 bits (milliseconds)
TMax     = RTT maximum (milliseconds)
TMin     = RTT minimum (milliseconds)

Entry Int BucI StartT     Pth Hop Comps OvrTh SumCmp     SumCmp2L   SumCmp2H   T
Max    TMin
```

Feature History and Information for Service Level Agreements

Release	Modification
Cisco IOS XE 3.3SE	This feature was introduced.



CHAPTER 6

Configuring SPAN and RSPAN

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SPAN and RSPAN

SPAN

- You can limit SPAN traffic to specific VLANs by using the **filter vlan** keyword. If a trunk port is being monitored, only traffic on the VLANs specified with this keyword is monitored. By default, all VLANs are monitored on a trunk port.

RSPAN

- We recommend that you configure an RSPAN VLAN before you configure an RSPAN source or a destination session.

Restrictions for SPAN and RSPAN

SPAN

The restrictions for SPAN are as follows:

- On each switch, you can configure 66 sessions. A maximum of 7 source sessions can be configured and the remaining sessions can be configured as RSPAN destinations sessions. A source session is either a local SPAN session or an RSPAN source session.
- For SPAN sources, you can monitor traffic for a single port or VLAN or a series or range of ports or VLANs for each session. You cannot mix source ports and source VLANs within a single SPAN session.
- The destination port cannot be a source port; a source port cannot be a destination port.
- You cannot have two SPAN sessions using the same destination port.
- When you configure a switch port as a SPAN destination port, it is no longer a normal switch port; only monitored traffic passes through the SPAN destination port.
- Entering SPAN configuration commands does not remove previously configured SPAN parameters. You must enter the **no monitor session** {*session_number* | **all** | **local** | **remote**} global configuration command to delete configured SPAN parameters.
- For local SPAN, outgoing packets through the SPAN destination port carry the original encapsulation headers—untagged, ISL, or IEEE 802.1Q—if the **encapsulation replicate** keywords are specified. If the keywords are not specified, the packets are sent in native form.
- You can configure a disabled port to be a source or destination port, but the SPAN function does not start until the destination port and at least one source port or source VLAN are enabled.
- You cannot mix source VLANs and filter VLANs within a single SPAN session.

Traffic monitoring in a SPAN session has the following restrictions:

- Sources can be ports or VLANs, but you cannot mix source ports and source VLANs in the same session.
- Wireshark does not capture egress packets when egress span is active.
- You can run both a local SPAN and an RSPAN source session in the same switch or switch stack. The switch or switch stack supports a total of 66 source and RSPAN destination sessions.
- You can configure two separate SPAN or RSPAN source sessions with separate or overlapping sets of SPAN source ports and VLANs. Both switched and routed ports can be configured as SPAN sources and destinations.
- You can have multiple destination ports in a SPAN session, but no more than 64 destination ports per switch stack.

- SPAN sessions do not interfere with the normal operation of the switch. However, an oversubscribed SPAN destination, for example, a 10-Mb/s port monitoring a 100-Mb/s port, can result in dropped or lost packets.
- When SPAN or RSPAN is enabled, each packet being monitored is sent twice, once as normal traffic and once as a monitored packet. Monitoring a large number of ports or VLANs could potentially generate large amounts of network traffic.
- You can configure SPAN sessions on disabled ports; however, a SPAN session does not become active unless you enable the destination port and at least one source port or VLAN for that session.
- The switch does not support a combination of local SPAN and RSPAN in a single session.
    - An RSPAN source session cannot have a local destination port.
    - An RSPAN destination session cannot have a local source port.
    - An RSPAN destination session and an RSPAN source session that are using the same RSPAN VLAN cannot run on the same switch or switch stack.

RSPAN

The restrictions for RSPAN are as follows:

- RSPAN does not support BPDU packet monitoring or other Layer 2 switch protocols.
- The RSPAN VLAN is configured only on trunk ports and not on access ports. To avoid unwanted traffic in RSPAN VLANs, make sure that the VLAN remote-span feature is supported in all the participating switches.
- RSPAN VLANs are included as sources for port-based RSPAN sessions when source trunk ports have active RSPAN VLANs. RSPAN VLANs can also be sources in SPAN sessions. However, since the switch does not monitor spanned traffic, it does not support egress spanning of packets on any RSPAN VLAN identified as the destination of an RSPAN source session on the switch.
- If you enable VTP and VTP pruning, RSPAN traffic is pruned in the trunks to prevent the unwanted flooding of RSPAN traffic across the network for VLAN IDs that are lower than 1005.
- To use RSPAN, the switch must be running the LAN Base image.

Information About SPAN and RSPAN

SPAN and RSPAN

You can analyze network traffic passing through ports or VLANs by using SPAN or RSPAN to send a copy of the traffic to another port on the switch or on another switch that has been connected to a network analyzer or other monitoring or security device. SPAN copies (or mirrors) traffic received or sent (or both) on source ports or source VLANs to a destination port for analysis. SPAN does not affect the switching of network traffic on the source ports or VLANs. You must dedicate the destination port for SPAN use. Except for traffic that is required for the SPAN or RSPAN session, destination ports do not receive or forward traffic.

Only traffic that enters or leaves source ports or traffic that enters or leaves source VLANs can be monitored by using SPAN; traffic routed to a source VLAN cannot be monitored. For example, if incoming traffic is

being monitored, traffic that gets routed from another VLAN to the source VLAN cannot be monitored; however, traffic that is received on the source VLAN and routed to another VLAN can be monitored.

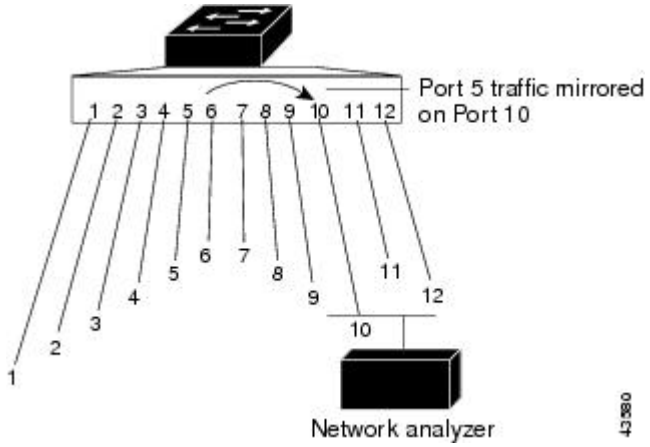
You can use the SPAN or RSPAN destination port to inject traffic from a network security device. For example, if you connect a Cisco Intrusion Detection System (IDS) sensor appliance to a destination port, the IDS device can send TCP reset packets to close down the TCP session of a suspected attacker.

Local SPAN

Local SPAN supports a SPAN session entirely within one switch; all source ports or source VLANs and destination ports are in the same switch or switch stack. Local SPAN copies traffic from one or more source ports in any VLAN or from one or more VLANs to a destination port for analysis.

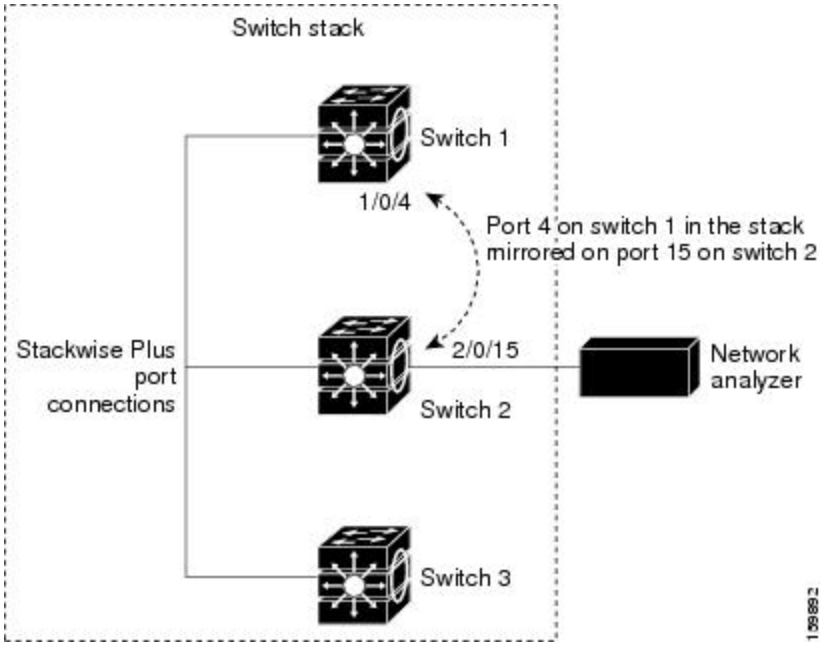
All traffic on port 5 (the source port) is mirrored to port 10 (the destination port). A network analyzer on port 10 receives all network traffic from port 5 without being physically attached to port 5.

Figure 6: Example of Local SPAN Configuration on a Single Device



This is an example of a local SPAN in a switch stack, where the source and destination ports reside on different stack members.

Figure 7: Example of Local SPAN Configuration on a Device Stack



Related Topics

Creating a Local SPAN Session, on page 103
Creating a Local SPAN Session and Configuring Incoming Traffic, on page 105
Example: Configuring Local SPAN, on page 122

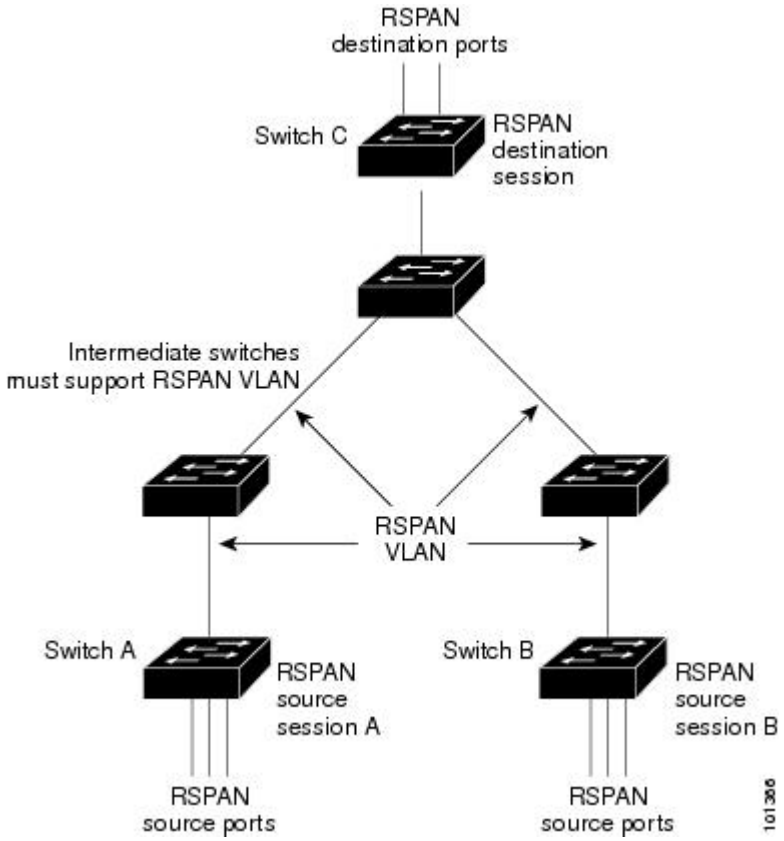
Remote SPAN

RSPAN supports source ports, source VLANs, and destination ports on different switches (or different switch stacks), enabling remote monitoring of multiple switches across your network.

The figure below shows source ports on Switch A and Switch B. The traffic for each RSPAN session is carried over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating switches. The RSPAN traffic from the source ports or VLANs is copied into the RSPAN VLAN and forwarded over trunk ports carrying the RSPAN VLAN to a destination session monitoring the RSPAN VLAN. Each RSPAN

source switch must have either ports or VLANs as RSPAN sources. The destination is always a physical port, as shown on Switch C in the figure.

Figure 8: Example of RSPAN Configuration



Related Topics

Creating an RSPAN Source Session, on page 110
Creating an RSPAN Destination Session, on page 114
Creating an RSPAN Destination Session and Configuring Incoming Traffic, on page 116
Examples: Creating an RSPAN VLAN, on page 123

SPAN and RSPAN Concepts and Terminology

- SPAN Sessions
- Monitored Traffic
- Source Ports
- Source VLANs
- VLAN Filtering
- Destination Port
- RSPAN VLAN

SPAN Sessions

SPAN sessions (local or remote) allow you to monitor traffic on one or more ports, or one or more VLANs, and send the monitored traffic to one or more destination ports.

A local SPAN session is an association of a destination port with source ports or source VLANs, all on a single network device. Local SPAN does not have separate source and destination sessions. Local SPAN sessions gather a set of ingress and egress packets specified by the user and form them into a stream of SPAN data, which is directed to the destination port.

RSPAN consists of at least one RSPAN source session, an RSPAN VLAN, and at least one RSPAN destination session. You separately configure RSPAN source sessions and RSPAN destination sessions on different network devices. To configure an RSPAN source session on a device, you associate a set of source ports or source VLANs with an RSPAN VLAN. The output of this session is the stream of SPAN packets that are sent to the RSPAN VLAN. To configure an RSPAN destination session on another device, you associate the destination port with the RSPAN VLAN. The destination session collects all RSPAN VLAN traffic and sends it out the RSPAN destination port.

An RSPAN source session is very similar to a local SPAN session, except for where the packet stream is directed. In an RSPAN source session, SPAN packets are relabeled with the RSPAN VLAN ID and directed over normal trunk ports to the destination switch.

An RSPAN destination session takes all packets received on the RSPAN VLAN, strips off the VLAN tagging, and presents them on the destination port. The session presents a copy of all RSPAN VLAN packets (except Layer 2 control packets) to the user for analysis.

A single RSPAN session with multiple source and destination ports can be in the same session but more than one source session with the source being the same remote vlan is not allowed.

Traffic monitoring in a SPAN session has these restrictions:

- Sources can be ports or VLANs, but you cannot mix source ports and source VLANs in the same session.
- The switch supports up to two local SPAN or RSPAN source sessions.
    - You can run both a local SPAN and an RSPAN source session in the same switch or switch stack. The switch or switch stack supports a total of 64 source and RSPAN destination sessions.
    - You can configure two separate SPAN or RSPAN source sessions with separate or overlapping sets of SPAN source ports and VLANs. Both switched and routed ports can be configured as SPAN sources and destinations.
- You can have multiple destination ports in a SPAN session, but no more than 64 destination ports per switch stack.
- SPAN sessions do not interfere with the normal operation of the switch. However, an oversubscribed SPAN destination, for example, a 10-Mb/s port monitoring a 100-Mb/s port, can result in dropped or lost packets.
- When SPAN or RSPAN is enabled, each packet being monitored is sent twice, once as normal traffic and once as a monitored packet. Therefore monitoring a large number of ports or VLANs could potentially generate large amounts of network traffic.
- You can configure SPAN sessions on disabled ports; however, a SPAN session does not become active unless you enable the destination port and at least one source port or VLAN for that session.
- The switch does not support a combination of local SPAN and RSPAN in a single session.

- An RSPAN source session cannot have a local destination port.
- An RSPAN destination session cannot have a local source port.
- An RSPAN destination session and an RSPAN source session that are using the same RSPAN VLAN cannot run on the same switch or switch stack.

Related Topics

Creating a Local SPAN Session, on page 103
Creating a Local SPAN Session and Configuring Incoming Traffic, on page 105
Example: Configuring Local SPAN, on page 122

Monitored Traffic

SPAN sessions can monitor these traffic types:

- Receive (Rx) SPAN—Receive (or ingress) SPAN monitors as much as possible all of the packets received by the source interface or VLAN before any modification or processing is performed by the switch. A copy of each packet received by the source is sent to the destination port for that SPAN session.

  Packets that are modified because of routing or Quality of Service (QoS)—for example, modified Differentiated Services Code Point (DSCP)—are copied before modification.

  Features that can cause a packet to be dropped during receive processing have no effect on ingress SPAN; the destination port receives a copy of the packet even if the actual incoming packet is dropped. These features include IP standard and extended input Access Control Lists (ACLs), ingress QoS policing, VLAN ACLs, and egress QoS policing.

- Transmit (Tx) SPAN—Transmit (or egress) SPAN monitors as much as possible all of the packets sent by the source interface after all modification and processing is performed by the switch. A copy of each packet sent by the source is sent to the destination port for that SPAN session. The copy is provided after the packet is modified.

  Packets that are modified because of routing (for example, with modified time-to-live (TTL), MAC address, or QoS values) are duplicated (with the modifications) at the destination port.

  Features that can cause a packet to be dropped during transmit processing also affect the duplicated copy for SPAN. These features include IP standard and extended output ACLs and egress QoS policing.

- Both—In a SPAN session, you can also monitor a port or VLAN for both received and sent packets. This is the default.

The default configuration for local SPAN session ports is to send all packets untagged. However, when you enter the **encapsulation replicate** keywords while configuring a destination port, these changes occur:

- Packets are sent on the destination port with the same encapsulation (untagged or IEEE 802.1Q) that they had on the source port.
- Packets of all types, including BPDU and Layer 2 protocol packets, are monitored.

Therefore, a local SPAN session with encapsulation replicate enabled can have a mixture of untagged and IEEE 802.1Q tagged packets appear on the destination port.

Switch congestion can cause packets to be dropped at ingress source ports, egress source ports, or SPAN destination ports. In general, these characteristics are independent of one another. For example:

- A packet might be forwarded normally but dropped from monitoring due to an oversubscribed SPAN destination port.
- An ingress packet might be dropped from normal forwarding, but still appear on the SPAN destination port.
- An egress packet dropped because of switch congestion is also dropped from egress SPAN.

In some SPAN configurations, multiple copies of the same source packet are sent to the SPAN destination port. For example, a bidirectional (both Rx and Tx) SPAN session is configured for the Rx monitor on port A and Tx monitor on port B. If a packet enters the switch through port A and is switched to port B, both incoming and outgoing packets are sent to the destination port. Both packets are the same unless a Layer 3 rewrite occurs, in which case the packets are different because of the packet modification.

Source Ports

A source port (also called a monitored port) is a switched or routed port that you monitor for network traffic analysis. In a local SPAN session or RSPAN source session, you can monitor source ports or VLANs for traffic in one or both directions. The switch supports any number of source ports (up to the maximum number of available ports on the switch) and any number of source VLANs (up to the maximum number of VLANs supported). However, the switch supports a maximum of (local or RSPAN) with source ports or VLANs. You cannot mix ports and VLANs in a single session.

A source port has these characteristics:

- It can be monitored in multiple SPAN sessions.
- Each source port can be configured with a direction (ingress, egress, or both) to monitor.
- It can be any port type (for example, EtherChannel, Gigabit Ethernet, and so forth).
- For EtherChannel sources, you can monitor traffic for the entire EtherChannel or individually on a physical port as it participates in the port channel.
- It can be an access port, trunk port, routed port, or voice VLAN port.
- It cannot be a destination port.
- Source ports can be in the same or different VLANs.
- You can monitor multiple source ports in a single session.

Source VLANs

VLAN-based SPAN (VSPAN) is the monitoring of the network traffic in one or more VLANs. The SPAN or RSPAN source interface in VSPAN is a VLAN ID, and traffic is monitored on all the ports for that VLAN.

VSPAN has these characteristics:

- All active ports in the source VLAN are included as source ports and can be monitored in either or both directions.
- On a given port, only traffic on the monitored VLAN is sent to the destination port.
- If a destination port belongs to a source VLAN, it is excluded from the source list and is not monitored.
- If ports are added to or removed from the source VLANs, the traffic on the source VLAN received by those ports is added to or removed from the sources being monitored.

- You cannot use filter VLANs in the same session with VLAN sources.
- You can monitor only Ethernet VLANs.

VLAN Filtering

When you monitor a trunk port as a source port, by default, all VLANs active on the trunk are monitored. You can limit SPAN traffic monitoring on trunk source ports to specific VLANs by using VLAN filtering.

- VLAN filtering applies only to trunk ports or to voice VLAN ports.
- VLAN filtering applies only to port-based sessions and is not allowed in sessions with VLAN sources.
- When a VLAN filter list is specified, only those VLANs in the list are monitored on trunk ports or on voice VLAN access ports.
- SPAN traffic coming from other port types is not affected by VLAN filtering; that is, all VLANs are allowed on other ports.
- VLAN filtering affects only traffic forwarded to the destination SPAN port and does not affect the switching of normal traffic.

Destination Port

Each local SPAN session or RSPAN destination session must have a destination port (also called a monitoring port) that receives a copy of traffic from the source ports or VLANs and sends the SPAN packets to the user, usually a network analyzer.

A destination port has these characteristics:

- For a local SPAN session, the destination port must reside on the same switch or switch stack as the source port. For an RSPAN session, it is located on the switch containing the RSPAN destination session. There is no destination port on a switch or switch stack running only an RSPAN source session.
- When a port is configured as a SPAN destination port, the configuration overwrites the original port configuration. When the SPAN destination configuration is removed, the port reverts to its previous configuration. If a configuration change is made to the port while it is acting as a SPAN destination port, the change does not take effect until the SPAN destination configuration had been removed.

Note When QoS is configured on the SPAN destination port, QoS takes effect immediately.

- If the port was in an EtherChannel group, it is removed from the group while it is a destination port. If it was a routed port, it is no longer a routed port.
- It can be any Ethernet physical port.
- It cannot be a secure port.
- It cannot be a source port.
- It can be an EtherChannel group (**ON** mode only).
- It cannot be a VLAN.

- It can participate in only one SPAN session at a time (a destination port in one SPAN session cannot be a destination port for a second SPAN session).
- When it is active, incoming traffic is disabled. The port does not transmit any traffic except that required for the SPAN session. Incoming traffic is never learned or forwarded on a destination port.
- If ingress traffic forwarding is enabled for a network security device, the destination port forwards traffic at Layer 2.
- It does not participate in any of the Layer 2 protocols (STP, VTP, CDP, DTP, PagP).
- A destination port that belongs to a source VLAN of any SPAN session is excluded from the source list and is not monitored.
- The maximum number of destination ports in a switch or switch stack is 64.

Local SPAN and RSPAN destination ports function differently with VLAN tagging and encapsulation:

- For local SPAN, if the **encapsulation replicate** keywords are specified for the destination port, these packets appear with the original encapsulation (untagged, ISL, or IEEE 802.1Q). If these keywords are not specified, packets appear in the untagged format. Therefore, the output of a local SPAN session with **encapsulation replicate** enabled can contain a mixture of untagged, ISL, or IEEE 802.1Q-tagged packets.
- For RSPAN, the original VLAN ID is lost because it is overwritten by the RSPAN VLAN identification. Therefore, all packets appear on the destination port as untagged.

RSPAN VLAN

The RSPAN VLAN carries SPAN traffic between RSPAN source and destination sessions. RSPAN VLAN has these special characteristics:

- All traffic in the RSPAN VLAN is always flooded.
- No MAC address learning occurs on the RSPAN VLAN.
- RSPAN VLAN traffic only flows on trunk ports.
- RSPAN VLANs must be configured in VLAN configuration mode by using the **remote-span** VLAN configuration mode command.
- STP can run on RSPAN VLAN trunks but not on SPAN destination ports.
- An RSPAN VLAN cannot be a private-VLAN primary or secondary VLAN.

For VLANs 1 to 1005 that are visible to VLAN Trunking Protocol (VTP), the VLAN ID and its associated RSPAN characteristic are propagated by VTP. If you assign an RSPAN VLAN ID in the extended VLAN range (1006 to 4094), you must manually configure all intermediate switches.

It is normal to have multiple RSPAN VLANs in a network at the same time with each RSPAN VLAN defining a network-wide RSPAN session. That is, multiple RSPAN source sessions anywhere in the network can contribute packets to the RSPAN session. It is also possible to have multiple RSPAN destination sessions throughout the network, monitoring the same RSPAN VLAN and presenting traffic to the user. The RSPAN VLAN ID separates the sessions.

Related Topics

Creating an RSPAN Source Session, on page 110
Creating an RSPAN Destination Session, on page 114

Creating an RSPAN Destination Session and Configuring Incoming Traffic, on page 116
Examples: Creating an RSPAN VLAN, on page 123

SPAN and RSPAN Interaction with Other Features

SPAN interacts with these features:

- Routing—SPAN does not monitor routed traffic. VSPAN only monitors traffic that enters or exits the switch, not traffic that is routed between VLANs. For example, if a VLAN is being Rx-monitored and the switch routes traffic from another VLAN to the monitored VLAN, that traffic is not monitored and not received on the SPAN destination port.
- STP—A destination port does not participate in STP while its SPAN or RSPAN session is active. The destination port can participate in STP after the SPAN or RSPAN session is disabled. On a source port, SPAN does not affect the STP status. STP can be active on trunk ports carrying an RSPAN VLAN.
- CDP—A SPAN destination port does not participate in CDP while the SPAN session is active. After the SPAN session is disabled, the port again participates in CDP.
- VTP—You can use VTP to prune an RSPAN VLAN between switches.
- VLAN and trunking—You can modify VLAN membership or trunk settings for source or destination ports at any time. However, changes in VLAN membership or trunk settings for a destination port do not take effect until you remove the SPAN destination configuration. Changes in VLAN membership or trunk settings for a source port immediately take effect, and the respective SPAN sessions automatically adjust accordingly.
- EtherChannel—You can configure an EtherChannel group as a source port or a SPAN destination port. When a group is configured as a SPAN source, the entire group is monitored.

  If a physical port is added to a monitored EtherChannel group, the new port is added to the SPAN source port list. If a port is removed from a monitored EtherChannel group, it is automatically removed from the source port list.

  A physical port that belongs to an EtherChannel group can be configured as a SPAN source port and still be a part of the EtherChannel. In this case, data from the physical port is monitored as it participates in the EtherChannel. However, if a physical port that belongs to an EtherChannel group is configured as a SPAN destination, it is removed from the group. After the port is removed from the SPAN session, it rejoins the EtherChannel group. Ports removed from an EtherChannel group remain members of the group, but they are in the inactive or suspended state.

  If a physical port that belongs to an EtherChannel group is a destination port and the EtherChannel group is a source, the port is removed from the EtherChannel group and from the list of monitored ports.
- Multicast traffic can be monitored. For egress and ingress port monitoring, only a single unedited packet is sent to the SPAN destination port. It does not reflect the number of times the multicast packet is sent.
- A private-VLAN port cannot be a SPAN destination port.
- A secure port cannot be a SPAN destination port.

  For SPAN sessions, do not enable port security on ports with monitored egress when ingress forwarding is enabled on the destination port. For RSPAN source sessions, do not enable port security on any ports with monitored egress.
- An IEEE 802.1x port can be a SPAN source port. You can enable IEEE 802.1x on a port that is a SPAN destination port; however, IEEE 802.1x is disabled until the port is removed as a SPAN destination.

For SPAN sessions, do not enable IEEE 802.1x on ports with monitored egress when ingress forwarding is enabled on the destination port. For RSPAN source sessions, do not enable IEEE 802.1x on any ports that are egress monitored.

SPAN and RSPAN and Device Stacks

Because the stack of switches represents one logical switch, local SPAN source ports and destination ports can be in different switches in the stack. Therefore, the addition or deletion of switches in the stack can affect a local SPAN session, as well as an RSPAN source or destination session. An active session can become inactive when a switch is removed from the stack or an inactive session can become active when a switch is added to the stack.

Flow-Based SPAN

You can control the type of network traffic to be monitored in SPAN or RSPAN sessions by using flow-based SPAN (FSPAN) or flow-based RSPAN (FRSPAN), which apply access control lists (ACLs) to the monitored traffic on the source ports. The FSPAN ACLs can be configured to filter IPv4, IPv6, and non-IP monitored traffic.

You apply an ACL to a SPAN session through the interface. It is applied to all the traffic that is monitored on all interfaces in the SPAN session.The packets that are permitted by this ACL are copied to the SPAN destination port. No other packets are copied to the SPAN destination port.

The original traffic continues to be forwarded, and any port, VLAN, and router ACLs attached are applied. The FSPAN ACL does not have any effect on the forwarding decisions. Similarly, the port, VLAN, and router ACLs do not have any effect on the traffic monitoring. If a security input ACL denies a packet and it is not forwarded, the packet is still copied to the SPAN destination ports if the FSPAN ACL permits it. But if the security output ACL denies a packet and it is not sent, it is not copied to the SPAN destination ports. However, if the security output ACL permits the packet to go out, it is only copied to the SPAN destination ports if the FSPAN ACL permits it. This is also true for an RSPAN session.

You can attach three types of FSPAN ACLs to the SPAN session:

- IPv4 FSPAN ACL— Filters only IPv4 packets.
- IPv6 FSPAN ACL— Filters only IPv6 packets.
- MAC FSPAN ACL— Filters only non-IP packets.

If a VLAN-based FSPAN session configured on a stack cannot fit in the hardware memory on one or more switches, it is treated as unloaded on those switches, and traffic meant for the FSPAN ACL and sourcing on that switch is not copied to the SPAN destination ports. The FSPAN ACL continues to be correctly applied, and traffic is copied to the SPAN destination ports on the switches where the FSPAN ACL fits in the hardware memory.

When an empty FSPAN ACL is attached, some hardware functions copy all traffic to the SPAN destination ports for that ACL. If sufficient hardware resources are not available, even an empty FSPAN ACL can be unloaded.

IPv4 and MAC FSPAN ACLs are supported on all feature sets. IPv6 FSPAN ACLs are supported only in the advanced IP Services feature set.

Related Topics

Configuring an FSPAN Session, on page 117
Configuring an FRSPAN Session, on page 120

Default SPAN and RSPAN Configuration

Table 13: Default SPAN and RSPAN Configuration

Feature	Default Setting
SPAN state (SPAN and RSPAN)	Disabled.
Source port traffic to monitor	Both received and sent traffic (**both**).
Encapsulation type (destination port)	Native form (untagged packets).
Ingress forwarding (destination port)	Disabled.
VLAN filtering	On a trunk interface used as a source port, all VLANs are monitored.
RSPAN VLANs	None configured.

Configuration Guidelines

SPAN Configuration Guidelines

- To remove a source or destination port or VLAN from the SPAN session, use the **no monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} global configuration command or the **no monitor session** *session_number* **destination interface** *interface-id* global configuration command. For destination interfaces, the **encapsulation** options are ignored with the **no** form of the command.
- To monitor all VLANs on the trunk port, use the **no monitor session** *session_number* **filter** global configuration command.

Related Topics

Creating a Local SPAN Session, on page 103
Creating a Local SPAN Session and Configuring Incoming Traffic, on page 105
Example: Configuring Local SPAN, on page 122

RSPAN Configuration Guidelines

- All the SPAN configuration guidelines apply to RSPAN.

- As RSPAN VLANs have special properties, you should reserve a few VLANs across your network for use as RSPAN VLANs; do not assign access ports to these VLANs.
- You can apply an output ACL to RSPAN traffic to selectively filter or monitor specific packets. Specify these ACLs on the RSPAN VLAN in the RSPAN source switches.
- For RSPAN configuration, you can distribute the source ports and the destination ports across multiple switches in your network.
- Access ports (including voice VLAN ports) on the RSPAN VLAN are put in the inactive state.
- You can configure any VLAN as an RSPAN VLAN as long as these conditions are met:
  - The same RSPAN VLAN is used for an RSPAN session in all the switches.
  - All participating switches support RSPAN.

Related Topics

Creating an RSPAN Source Session, on page 110
Creating an RSPAN Destination Session, on page 114
Creating an RSPAN Destination Session and Configuring Incoming Traffic, on page 116
Examples: Creating an RSPAN VLAN, on page 123

FSPAN and FRSPAN Configuration Guidelines

- When at least one FSPAN ACL is attached, FSPAN is enabled.
- When you attach at least one FSPAN ACL that is not empty to a SPAN session, and you have not attached one or more of the other FSPAN ACLs (for instance, you have attached an IPv4 ACL that is not empty, and have not attached IPv6 and MAC ACLs), FSPAN blocks the traffic that would have been filtered by the unattached ACLs. Therefore, this traffic is not monitored.

Related Topics

Configuring an FSPAN Session, on page 117
Configuring an FRSPAN Session, on page 120

How to Configure SPAN and RSPAN

Creating a Local SPAN Session

Beginning in privileged EXEC mode, follow these steps to create a SPAN session and specify the source (monitored) ports or VLANs and the destination (monitoring) ports.

SUMMARY STEPS

1. **configure terminal**
2. **no monitor session** {*session_number* | **all** | **local** | **remote**}
3. **monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**]
4. **monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Switch(config)# no monitor session all`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all—**Removes all SPAN sessions. • **local—**Removes all local sessions. • **remote—**Removes all remote SPAN sessions.
Step 3	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] **Example:** `Switch(config)# monitor session 1 source interface gigabitethernet1/0/1`	Specifies the SPAN session and the source port (monitored port). • For *session_number*, the range is 1 to 66. • For *interface-id*, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** *port-channel-number*). Valid port-channel numbers are 1 to 48. • For *vlan-id*, specify the source VLAN to monitor. The range is 1 to 4094 (excluding the RSPAN VLAN). **Note** A single session can include multiple sources (ports or VLANs) defined in a series of commands, but you cannot combine source ports and source VLANs in one session. • (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) **both** | **rx** | **tx—**Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic. ◦ **both—**Monitors both received and sent traffic. ◦ **rx—**Monitors received traffic. ◦ **tx—**Monitors sent traffic.

	Command or Action	Purpose
		Note You can use the **monitor session** *session_number* **source** command multiple times to configure multiple source ports.
Step 4	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]} **Example:** `Switch(config)# monitor session 1 destination interface gigabitethernet1/0/2 encapsulation replicate`	Specifies the SPAN session and the destination port (monitoring port). **Note** For local SPAN, you must use the same session number for the source and destination interfaces. • For *session_number*, specify the session number entered in step 3. • For *interface-id*, specify the destination port. The destination interface must be a physical port; it cannot be an EtherChannel, and it cannot be a VLAN. • (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged). **Note** You can use **monitor session** *session_number* **destination** command multiple times to configure multiple destination ports.
Step 5	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Related Topics

Local SPAN, on page 92
SPAN Sessions, on page 95
SPAN Configuration Guidelines, on page 102

Creating a Local SPAN Session and Configuring Incoming Traffic

Beginning in privileged EXEC mode, follow these steps to create a SPAN session, to specify the source ports or VLANs and the destination ports, and to enable incoming traffic on the destination port for a network security device (such as a Cisco IDS Sensor Appliance).

SUMMARY STEPS

1. **configure terminal**
2. **no monitor session** {*session_number* | **all** | **local** | **remote**}
3. **monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**]
4. **monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**] [**ingress** {**dot1q vlan** *vlan-id* **| isl | untagged vlan** *vlan-id* **| vlan** *vlan-id*}]}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Switch(config)# no monitor session all`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all—**Removes all SPAN sessions. • **local—**Removes all local sessions. • **remote—**Removes all remote SPAN sessions.
Step 3	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] **Example:** `Switch(config)# monitor session 2 source gigabitethernet1/0/1 rx`	Specifies the SPAN session and the source port (monitored port).
Step 4	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**] [**ingress** {**dot1q vlan** *vlan-id* **| isl | untagged vlan** *vlan-id* **| vlan** *vlan-id*}]} **Example:** `Switch(config)# monitor session 2 destination interface gigabitethernet1/0/2 encapsulation replicate ingress dot1q vlan 6`	Specifies the SPAN session, the destination port, the packet encapsulation, and the ingress VLAN and encapsulation. • For *session_number*, specify the session number entered in Step 3. • For *interface-id*, specify the destination port. The destination interface must be a physical port; it cannot be an EtherChannel, and it cannot be a VLAN. • (Optional) [**,** | **-**]—Specifies a series or range of interfaces. Enter a space before and after the comma or hyphen. • (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged).

	Command or Action	Purpose
		• **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type: ◦ **dot1q vlan** *vlan-id*—Accepts incoming packets with IEEE 802.1Q encapsulation with the specified VLAN as the default VLAN. ◦ **isl**—Forwards ingress packets with ISL encapsulation. ◦ **untagged vlan** *vlan-id* or **vlan** *vlan-id*—Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN. • **dot1q vlan** *vlan-id*—Accepts incoming packets with IEEE 802.1Q encapsulation with the specified VLAN as the default VLAN. • **isl**—Forwards ingress packets with ISL encapsulation. • **untagged vlan** *vlan-id* or **vlan** *vlan-id*—Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.
Step 5	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Related Topics

Local SPAN, on page 92
SPAN Sessions, on page 95
SPAN Configuration Guidelines, on page 102
Example: Configuring Local SPAN, on page 122

Specifying VLANs to Filter

Beginning in privileged EXEC mode, follow these steps to limit SPAN source traffic to specific VLANs.

SUMMARY STEPS

1. **configure terminal**
2. **no monitor session** {*session_number* | **all** | **local** | **remote**}
3. **monitor session** *session_number* **source interface** *interface-id*
4. **monitor session** *session_number* **filter vlan** *vlan-id* [**,** | **-**]
5. **monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Switch(config)# no monitor session all`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all—**Removes all SPAN sessions. • **local—**Removes all local sessions. • **remote—**Removes all remote SPAN sessions.
Step 3	**monitor session** *session_number* **source interface** *interface-id* **Example:** `Switch(config)# monitor session 2 source interface gigabitethernet1/0/2 rx`	Specifies the characteristics of the source port (monitored port) and SPAN session. • For *session_number*, the range is 1 to 66. • For *interface-id*, specify the source port to monitor. The interface specified must already be configured as a trunk port.
Step 4	**monitor session** *session_number* **filter vlan** *vlan-id* [**,** | **-**] **Example:** `Switch(config)# monitor session 2 filter vlan 1 - 5 , 9`	Limits the SPAN source traffic to specific VLANs. • For *session_number*, enter the session number specified in Step 3. • For *vlan-id*, the range is 1 to 4094. • (Optional) Use a comma (**,**) to specify a series of VLANs, or use a hyphen (**-**) to specify a range of VLANs. Enter a space before and after the comma; enter a space before and after the hyphen.
Step 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]}	Specifies the SPAN session and the destination port (monitoring port). • For *session_number*, specify the session number entered in Step 3.

	Command or Action	Purpose
	Example: `Switch(config)# monitor session 2 destination interface gigabitethernet1/0/1`	• For *interface-id*, specify the destination port. The destination interface must be a physical port; it cannot be an EtherChannel, and it cannot be a VLAN. • (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged).
Step 6	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Configuring a VLAN as an RSPAN VLAN

Beginning in privileged EXEC mode, follow these steps to create a new VLAN, then configure it to be the RSPAN VLAN for the RSPAN session.

SUMMARY STEPS

1. **configure terminal**
2. **vlan** *vlan-id*
3. **remote-span**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.

	Command or Action	Purpose
Step 2	**vlan** *vlan-id* **Example:** `Switch(config)# vlan 100`	Enters a VLAN ID to create a VLAN, or enters the VLAN ID of an existing VLAN, and enters VLAN configuration mode. The range is 2 to 1001 and 1006 to 4094. The RSPAN VLAN cannot be VLAN 1 (the default VLAN) or VLAN IDs 1002 through 1005 (reserved for Token Ring and FDDI VLANs).
Step 3	**remote-span** **Example:** `Switch(config-vlan)# remote-span`	Configures the VLAN as an RSPAN VLAN.
Step 4	**end** **Example:** `Switch(config-vlan)# end`	Returns to privileged EXEC mode.

What to Do Next

You must create the RSPAN VLAN in all switches that will participate in RSPAN. If the RSPAN VLAN-ID is in the normal range (lower than 1005) and VTP is enabled in the network, you can create the RSPAN VLAN in one switch, and VTP propagates it to the other switches in the VTP domain. For extended-range VLANs (greater than 1005), you must configure RSPAN VLAN on both source and destination switches and any intermediate switches.

Use VTP pruning to get an efficient flow of RSPAN traffic, or manually delete the RSPAN VLAN from all trunks that do not need to carry the RSPAN traffic.

To remove the remote SPAN characteristic from a VLAN and convert it back to a normal VLAN, use the **no remote-span** VLAN configuration command.

To remove a source port or VLAN from the SPAN session, use the **no monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} global configuration command. To remove the RSPAN VLAN from the session, use the **no monitor session** *session_number* **destination remote vlan** *vlan-id*.

Creating an RSPAN Source Session

Beginning in privileged EXEC mode, follow these steps to create and start an RSPAN source session and to specify the monitored source and the destination RSPAN VLAN.

SUMMARY STEPS

1. **configure terminal**
2. **no monitor session** {*session_number* | **all** | **local** | **remote**}
3. **monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**]
4. **monitor session** *session_number* **destination remote vlan** *vlan-id*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Switch(config)# no monitor session 1`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all—**Removes all SPAN sessions. • **local—**Removes all local sessions. • **remote—**Removes all remote SPAN sessions.
Step 3	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] **Example:** `Switch(config)# monitor session 1 source interface gigabitethernet1/0/1 tx`	Specifies the RSPAN session and the source port (monitored port). • For *session_number*, the range is 1 to 66. • Enter a source port or source VLAN for the RSPAN session: ◦ For *interface-id*, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** *port-channel-number*). Valid port-channel numbers are 1 to 48. ◦ For *vlan-id*, specifies the source VLAN to monitor. The range is 1 to 4094 (excluding the RSPAN VLAN). A single session can include multiple sources (ports or VLANs), defined in a series of commands, but you cannot combine source ports and source VLANs in one session. • (Optional) [**,** | **-**]—Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) **both** | **rx** | **tx—**Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic.

	Command or Action	Purpose
		◦ **both—**Monitors both received and sent traffic. ◦ **rx—**Monitors received traffic. ◦ **tx—**Monitors sent traffic.
Step 4	**monitor session** *session_number* **destination remote vlan** *vlan-id* **Example:** `Switch(config)# monitor session 1 destination remote vlan 100`	Specifies the RSPAN session, the destination RSPAN VLAN, and the destination-port group. • For *session_number*, enter the number defined in Step 3. • For *vlan-id*, specify the source RSPAN VLAN to monitor.
Step 5	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Related Topics

Remote SPAN, on page 93
RSPAN VLAN, on page 99
RSPAN Configuration Guidelines, on page 102

Specifying VLANs to Filter

Beginning in privileged EXEC mode, follow these steps to configure the RSPAN source session to limit RSPAN source traffic to specific VLANs.

SUMMARY STEPS

1. **configure terminal**
2. **no monitor session** {*session_number* | **all** | **local** | **remote**}
3. **monitor session** *session_number* **source interface** *interface-id*
4. **monitor session** *session_number* **filter vlan** *vlan-id* [**,** | **-**]
5. **monitor session** *session_number* **destination remote vlan** *vlan-id*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Switch(config)# no monitor session 2`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all—**Removes all SPAN sessions. • **local—**Removes all local sessions. • **remote—**Removes all remote SPAN sessions.
Step 3	**monitor session** *session_number* **source interface** *interface-id* **Example:** `Switch(config)# monitor session 2 source interface gigabitethernet1/0/2 rx`	Specifies the characteristics of the source port (monitored port) and SPAN session. • For *session_number*, the range is 1 to 66. • For *interface-id*, specify the source port to monitor. The interface specified must already be configured as a trunk port.
Step 4	**monitor session** *session_number* **filter vlan** *vlan-id* [**,** | **-**] **Example:** `Switch(config)# monitor session 2 filter vlan 1 - 5 , 9`	Limits the SPAN source traffic to specific VLANs. • For *session_number*, enter the session number specified in step 3. • For *vlan-id*, the range is 1 to 4094. • (Optional) **,** | **-** Use a comma (**,**) to specify a series of VLANs or use a hyphen (**-**) to specify a range of VLANs. Enter a space before and after the comma; enter a space before and after the hyphen.
Step 5	**monitor session** *session_number* **destination remote vlan** *vlan-id* **Example:** `Switch(config)# monitor session 2 destination remote vlan 902`	Specifies the RSPAN session and the destination remote VLAN (RSPAN VLAN). • For *session_number*, enter the session number specified in Step 3. • For *vlan-id*, specify the RSPAN VLAN to carry the monitored traffic to the destination port.
Step 6	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Creating an RSPAN Destination Session

You configure an RSPAN destination session on a different switch or switch stack; that is, not the switch or switch stack on which the source session was configured.

Beginning in privileged EXEC mode, follow these steps to define the RSPAN VLAN on that switch, to create an RSPAN destination session, and to specify the source RSPAN VLAN and the destination port.

SUMMARY STEPS

1. **configure terminal**
2. **vlan** *vlan-id*
3. **remote-span**
4. **exit**
5. **no monitor session** {*session_number* | **all** | **local** | **remote**}
6. **monitor session** *session_number* **source remote vlan** *vlan-id*
7. **monitor session** *session_number* **destination interface** *interface-id*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**vlan** *vlan-id* **Example:** `Switch(config)# vlan 901`	Specifies the VLAN ID of the RSPAN VLAN created from the source switch, and enters VLAN configuration mode. If both switches are participating in VTP and the RSPAN VLAN ID is from 2 to 1005, Steps 2 through 4 are not required because the RSPAN VLAN ID is propagated through the VTP network.
Step 3	**remote-span** **Example:** `Switch(config-vlan)# remote-span`	Identifies the VLAN as the RSPAN VLAN.
Step 4	**exit** **Example:** `Switch(config-vlan)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 5	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Switch(config)# no monitor session 1`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all—**Removes all SPAN sessions. • **local—**Removes all local sessions. • **remote—**Removes all remote SPAN sessions.
Step 6	**monitor session** *session_number* **source remote vlan** *vlan-id* **Example:** `Switch(config)# monitor session 1 source remote vlan 901`	Specifies the RSPAN session and the source RSPAN VLAN. • For *session_number*, the range is 1 to 66. • For *vlan-id*, specify the source RSPAN VLAN to monitor.
Step 7	**monitor session** *session_number* **destination interface** *interface-id* **Example:** `Switch(config)# monitor session 1 destination interface gigabitethernet2/0/1`	Specifies the RSPAN session and the destination interface. • For *session_number*, enter the number defined in Step 6. In an RSPAN destination session, you must use the same session number for the source RSPAN VLAN and the destination port. • For *interface-id*, specify the destination interface. The destination interface must be a physical interface. • Though visible in the command-line help string, **encapsulation replicate** is not supported for RSPAN. The original VLAN ID is overwritten by the RSPAN VLAN ID, and all packets appear on the destination port as untagged.
Step 8	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Related Topics

Remote SPAN, on page 93
RSPAN VLAN, on page 99
RSPAN Configuration Guidelines, on page 102

Creating an RSPAN Destination Session and Configuring Incoming Traffic

Beginning in privileged EXEC mode, follow these steps to create an RSPAN destination session, to specify the source RSPAN VLAN and the destination port, and to enable incoming traffic on the destination port for a network security device (such as a Cisco IDS Sensor Appliance).

SUMMARY STEPS

1. **configure terminal**
2. **no monitor session** {*session_number* | **all** | **local** | **remote**}
3. **monitor session** *session_number* **source remote vlan** *vlan-id*
4. **monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**ingress** {**dot1q vlan** *vlan-id* | **isl** | **untagged vlan** *vlan-id* | **vlan** *vlan-id*}]}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters the global configuration mode.
Step 2	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Switch(config)# no monitor session 2`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all—**Removes all SPAN sessions. • **local—**Removes all local sessions. • **remote—**Removes all remote SPAN sessions.
Step 3	**monitor session** *session_number* **source remote vlan** *vlan-id* **Example:** `Switch(config)# monitor session 2 source remote vlan 901`	Specifies the RSPAN session and the source RSPAN VLAN. • For *session_number*, the range is 1 to 66. • For *vlan-id*, specify the source RSPAN VLAN to monitor.
Step 4	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**ingress** {**dot1q vlan** *vlan-id* | **isl** | **untagged vlan** *vlan-id* | **vlan** *vlan-id*}]} **Example:** `Switch(config)# monitor session 2`	Specifies the SPAN session, the destination port, the packet encapsulation, and the incoming VLAN and encapsulation. • For *session_number*, enter the number defined in Step 4. In an RSPAN destination session, you must use the same session number for the source RSPAN VLAN and the destination port.

	Command or Action	Purpose
	`destination interface gigabitethernet1/0/2 ingress vlan 6`	• For *interface-id*, specify the destination interface. The destination interface must be a physical interface. • Though visible in the command-line help string, **encapsulation replicate** is not supported for RSPAN. The original VLAN ID is overwritten by the RSPAN VLAN ID, and all packets appear on the destination port as untagged. • (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • Enter **ingress** with additional keywords to enable forwarding of incoming traffic on the destination port and to specify the encapsulation type: ◦ **dot1q vlan** *vlan-id*—Forwards incoming packets with IEEE 802.1Q encapsulation with the specified VLAN as the default VLAN. ◦ **isl**—Forwards ingress packets with ISL encapsulation. ◦ **untagged vlan** *vlan-id* or **vlan** *vlan-id*—Forwards incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.
Step 5	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Related Topics

Remote SPAN, on page 93
RSPAN VLAN, on page 99
RSPAN Configuration Guidelines, on page 102
Examples: Creating an RSPAN VLAN, on page 123

Configuring an FSPAN Session

Beginning in privileged EXEC mode, follow these steps to create a SPAN session, specify the source (monitored) ports or VLANs and the destination (monitoring) ports, and configure FSPAN for the session.

SUMMARY STEPS

1. **configure terminal**
2. **no monitor session** {*session_number* | **all** | **local** | **remote**}
3. **monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**]
4. **monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]}
5. **monitor session** *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Switch(config)# no monitor session 2`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all—**Removes all SPAN sessions. • **local—**Removes all local sessions. • **remote—**Removes all remote SPAN sessions.
Step 3	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] **Example:** `Switch(config)# monitor session 2 source interface gigabitethernet1/0/1`	Specifies the SPAN session and the source port (monitored port). • For *session_number*, the range is 1 to 66. • For *interface-id*, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** *port-channel-number*). Valid port-channel numbers are 1 to 48. • For *vlan-id*, specify the source VLAN to monitor. The range is 1 to 4094 (excluding the RSPAN VLAN). **Note** A single session can include multiple sources (ports or VLANs) defined in a series of commands, but you cannot combine source ports and source VLANs in one session. • (Optional) [**,** | **-**]—Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) [**both** | **rx** | **tx**]—Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the SPAN monitors both sent and received traffic. ◦ **both—**Monitors both sent and received traffic. This is the default. ◦ **rx—**Monitors received traffic.

	Command or Action	Purpose
		◦ **tx—**Monitors sent traffic. **Note** You can use the **monitor session** *session_number* **source** command multiple times to configure multiple source ports.
Step 4	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]} **Example:** `Switch(config)# monitor session 2 destination interface gigabitethernet1/0/2 encapsulation replicate`	Specifies the SPAN session and the destination port (monitoring port). • For *session_number*, specify the session number entered in Step 3. • For **destination**, specify the following parameters: ◦ For *interface-id*, specify the destination port. The destination interface must be a physical port; it cannot be an EtherChannel, and it cannot be a VLAN. ◦ (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. ◦ (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged). **Note** For local SPAN, you must use the same session number for the source and destination interfaces. You can use **monitor session** *session_number* **destination** command multiple times to configure multiple destination ports.
Step 5	**monitor session** *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*} **Example:** `Switch(config)# monitor session 2 filter ipv6 access-group 4`	Specifies the SPAN session, the types of packets to filter, and the ACLs to use in an FSPAN session. • For *session_number*, specify the session number entered in Step 3. • For *access-list-number*, specify the ACL number that you want to use to filter traffic. • For *name*, specify the ACL name that you want to use to filter traffic.
Step 6	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Related Topics

Flow-Based SPAN, on page 101
FSPAN and FRSPAN Configuration Guidelines, on page 103

Configuring an FRSPAN Session

Beginning in privileged EXEC mode, follow these steps to start an RSPAN source session, specify the monitored source and the destination RSPAN VLAN, and configure FRSPAN for the session.

SUMMARY STEPS

1. **configure terminal**
2. **no monitor session** {*session_number* | **all** | **local** | **remote**}
3. **monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**]
4. **monitor session** *session_number* **destination remote vlan** *vlan-id*
5. **vlan** *vlan-id*
6. **remote-span**
7. **exit**
8. **monitor session** *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*}
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**no monitor session** {*session_number* | **all** | **local** | **remote**} **Example:** `Switch(config)# no monitor session 2`	Removes any existing SPAN configuration for the session. • For *session_number*, the range is 1 to 66. • **all—**Removes all SPAN sessions. • **local—**Removes all local sessions. • **remote—**Removes all remote SPAN sessions.
Step 3	**monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] **Example:** `Switch(config)# monitor session 2 source interface gigabitethernet1/0/1`	Specifies the SPAN session and the source port (monitored port). • For *session_number*, the range is 1 to 66. • For *interface-id*, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** *port-channel-number*). Valid port-channel numbers are 1 to 48. • For *vlan-id*, specify the source VLAN to monitor. The range is 1 to 4094 (excluding the RSPAN VLAN). **Note** A single session can include multiple sources (ports or VLANs) defined in a series of commands, but you cannot combine source ports and source VLANs in one session.

	Command or Action	Purpose
		• (Optional) [**,** | **-**]—Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) [**both** | **rx** | **tx**]—Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the SPAN monitors both sent and received traffic. • **both**—Monitors both sent and received traffic. This is the default. • **rx**—Monitors received traffic. • **tx**—Monitors sent traffic. **Note** You can use the **monitor session** *session_number* **source** command multiple times to configure multiple source ports.
Step 4	**monitor session** *session_number* **destination remote vlan** *vlan-id* **Example:** `Switch(config)# monitor session 2 destination remote vlan 5`	Specifies the RSPAN session and the destination RSPAN VLAN. • For *session_number*, enter the number defined in Step 3. • For *vlan-id*, specify the destination RSPAN VLAN to monitor.
Step 5	**vlan** *vlan-id* **Example:** `Switch(config)# vlan 10`	Enters the VLAN configuration mode. For *vlan-id*, specify the source RSPAN VLAN to monitor.
Step 6	**remote-span** **Example:** `Switch(config-vlan)# remote-span`	Specifies that the VLAN you specified in Step 5 is part of the RSPAN VLAN.
Step 7	**exit** **Example:** `Switch(config-vlan)# exit`	Returns to global configuration mode.
Step 8	**monitor session** *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*} **Example:** `Switch(config)# monitor session 2 filter ip access-group 7`	Specifies the RSPAN session, the types of packets to filter, and the ACLs to use in an FRSPAN session. • For *session_number*, specify the session number entered in Step 3. • For *access-list-number*, specify the ACL number that you want to use to filter traffic. • For *name*, specify the ACL name that you want to use to filter traffic.

	Command or Action	Purpose
Step 9	**end** **Example:** `Switch(config)# end`	Returns to privileged EXEC mode.

Related Topics

Flow-Based SPAN, on page 101
FSPAN and FRSPAN Configuration Guidelines, on page 103

Monitoring SPAN and RSPAN Operations

The following table describes the command used to display SPAN and RSPAN operations configuration and results to monitor operations:

Table 14: Monitoring SPAN and RSPAN Operations

Command	Purpose
show monitor	Displays the current SPAN, RSPAN, FSPAN, or FRSPAN configuration.

SPAN and RSPAN Configuration Examples

Example: Configuring Local SPAN

This example shows how to set up SPAN session 1 for monitoring source port traffic to a destination port. First, any existing SPAN configuration for session 1 is deleted, and then bidirectional traffic is mirrored from source Gigabit Ethernet port 1 to destination Gigabit Ethernet port 2, retaining the encapsulation method.

```
Switch(config)# no monitor session 1
Switch(config)# monitor session 1 source interface gigabitethernet1/0/1
Switch(config)# monitor session 1 destination interface gigabitethernet1/0/2
encapsulation replicate
Switch(config)# end
```

This example shows how to remove port 1 as a SPAN source for SPAN session 1:

```
Switch(config)# no monitor session 1 source interface gigabitethernet1/0/1
Switch(config)# end
```

This example shows how to disable received traffic monitoring on port 1, which was configured for bidirectional monitoring:

```
Switch(config)# no monitor session 1 source interface gigabitethernet1/0/1 rx
```

The monitoring of traffic received on port 1 is disabled, but traffic sent from this port continues to be monitored.

This example shows how to remove any existing configuration on SPAN session 2, configure SPAN session 2 to monitor received traffic on all ports belonging to VLANs 1 through 3, and send it to destination Gigabit Ethernet port 2. The configuration is then modified to also monitor all traffic on all ports belonging to VLAN 10.

```
Switch(config)# no monitor session 2
Switch(config)# monitor session 2 source vlan 1 - 3 rx
Switch(config)# monitor session 2 destination interface gigabitethernet1/0/2
Switch(config)# monitor session 2 source vlan 10
Switch(config)# end
```

This example shows how to remove any existing configuration on SPAN session 2, configure SPAN session 2 to monitor received traffic on Gigabit Ethernet source port 1, and send it to destination Gigabit Ethernet port 2 with the same egress encapsulation type as the source port, and to enable ingress forwarding with IEEE 802.1Q encapsulation and VLAN 6 as the default ingress VLAN:

```
Switch(config)# no monitor session 2
Switch(config)# monitor session 2 source gigabitethernet1/0/1 rx
Switch(config)# monitor session 2 destination interface gigabitethernet1/0/2 encapsulation
 replicate ingress dot1q vlan 6
Switch(config)# end
```

This example shows how to remove any existing configuration on SPAN session 2, configure SPAN session 2 to monitor traffic received on Gigabit Ethernet trunk port 2, and send traffic for only VLANs 1 through 5 and VLAN 9 to destination Gigabit Ethernet port 1:

```
Switch(config)# no monitor session 2
Switch(config)# monitor session 2 source interface gigabitethernet1/0/2 rx
Switch(config)# monitor session 2 filter vlan 1 - 5 , 9
Switch(config)# monitor session 2 destination interface gigabitethernet1/0/1
Switch(config)# end
```

Related Topics

Creating a Local SPAN Session and Configuring Incoming Traffic, on page 105
Local SPAN, on page 92
SPAN Sessions, on page 95
SPAN Configuration Guidelines, on page 102

Examples: Creating an RSPAN VLAN

This example shows how to create the RSPAN VLAN 901:

```
Switch(config)# vlan 901
Switch(config-vlan)# remote span
Switch(config-vlan)# end
```

This example shows how to remove any existing RSPAN configuration for session 1, configure RSPAN session 1 to monitor multiple source interfaces, and configure the destination as RSPAN VLAN 901:

```
Switch(config)# no monitor session 1
```

```
Switch(config)# monitor session 1 source interface gigabitethernet1/0/1 tx
Switch(config)# monitor session 1 source interface gigabitethernet1/0/2 rx
Switch(config)# monitor session 1 source interface port-channel 2
Switch(config)# monitor session 1 destination remote vlan 901
Switch(config)# end
```

This example shows how to remove any existing configuration on RSPAN session 2, configure RSPAN session 2 to monitor traffic received on trunk port 2, and send traffic for only VLANs 1 through 5 and 9 to destination RSPAN VLAN 902:

```
Switch(config)# no monitor session 2
Switch(config)# monitor session 2 source interface gigabitethernet1/0/2 rx
Switch(config)# monitor session 2 filter vlan 1 - 5 , 9
Switch(config)# monitor session 2 destination remote vlan 902
Switch(config)# end
```

This example shows how to configure VLAN 901 as the source remote VLAN and port 1 as the destination interface:

```
Switch(config)# monitor session 1 source remote vlan 901
Switch(config)# monitor session 1 destination interface gigabitethernet2/0/1
Switch(config)# end
```

This example shows how to configure VLAN 901 as the source remote VLAN in RSPAN session 2, to configure Gigabit Ethernet source port 2 as the destination interface, and to enable forwarding of incoming traffic on the interface with VLAN 6 as the default receiving VLAN:

```
Switch(config)# monitor session 2 source remote vlan 901
Switch(config)# monitor session 2 destination interface gigabitethernet1/0/2 ingress vlan 6
Switch(config)# end
```

Related Topics

Creating an RSPAN Destination Session and Configuring Incoming Traffic, on page 116
Remote SPAN, on page 93
RSPAN VLAN, on page 99
RSPAN Configuration Guidelines, on page 102

Additional References

Related Documents

Related Topic	Document Title

Error Message Decoder

Description	Link
To help you research and resolve system error messages in this release, use the Error Message Decoder tool.	https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

Standards and RFCs

Standard/RFC	Title

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

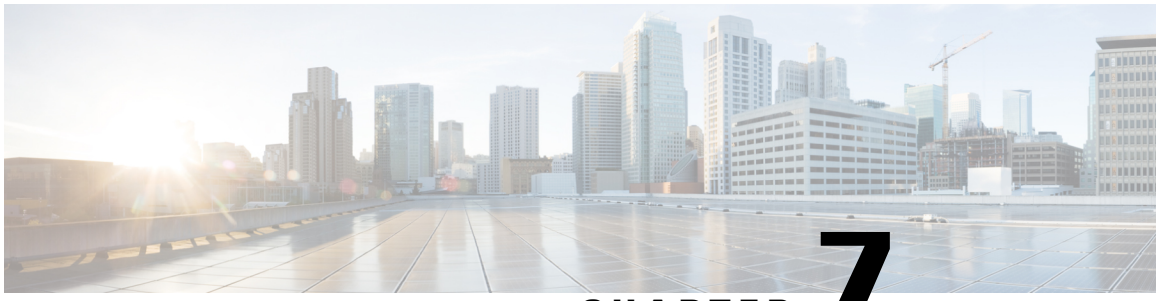
Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History and Information for SPAN and RSPAN

Release	Modification
Cisco IOS XE 3.3SE	Switch Port Analyzer (SPAN): Allows monitoring of switch traffic on a port or VLAN using a sniffer/analyzer or RMON probe. This feature was introduced.

Release	Modification
Cisco IOS XE 3.3SE	Flow-based Switch Port Analyzer (SPAN): Provides a method to capture only required data between end hosts by using specified filters. The filters are defined in terms of access lists that limit IPv4, IPv6 or IPv4 + IPv6, or non-IP traffic (MAC) between specified source and destination addresses. This feature was introduced.
Cisco IOS XE 3.3SE	SPAN destination port support on EtherChannels: Provides the ability to configure a SPAN destination port on an EtherChannel. This feature was introduced.
Cisco IOS XE 3.3SE	Switch Port Analyzer (SPAN) - distributed egress SPAN: Provides distributed egress SPAN functionality onto line cards in conjunction with ingress SPAN already been distributed to line cards. By distributing egress SPAN functionalities onto line cards, the performance of the system is improved. This feature was introduced.



CHAPTER 7

Configuring Wireshark

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Wireshark

- Wireshark is supported on Supervisor Engine 7-E, Supervisor Engine 7L-E, Catalyst 3850, Catalyst 3650, Wireless LAN Controller 5700 Series, Catalyst 4500X-16, and Catalyst 4500X-32.

Restrictions for Wireshark

- Starting in Cisco IOS Release XE 3.3.0(SE), global packet capture on Wireshark is not supported.

- Capture filters are not supported.
- The CLI for configuring Wireshark requires that the feature be executed only from EXEC mode. Actions that usually occur in configuration submode (such as defining capture points), are handled at the EXEC mode instead. All key commands are not NVGEN'd and are not synchronized to the standby supervisor in NSF and SSO scenarios.
- Packets captured in the output direction of an interface might not reflect the changes made by switch rewrite (includes TTL, VLAN tag, CoS, checksum, MAC addresses, DSCP, precedent, UP, etc.).
- Limiting circular file storage by file size is not supported.

Wireless Packet Capture

- The only form of wireless capture is a CAPWAP tunnel capture.
- When capturing CAPWAP tunnels, no other interface types can be used as attachment points on the same capture point.
- Capturing multiple CAPWAP tunnels is supported.
- Core filters are not applied and should be omitted when capturing a CAPWAP tunnel.
- To capture a CAPWAP data tunnel, each CAPWAP tunnel is mapped to a physical port and an appropriate ACL will be applied to filter the traffic.
- To capture a CAPWAP non-data tunnel, the switch is set to capture traffic on all ports and apply an appropriate ACL to filter the traffic.

Configuration Limitations

- Multiple capture points can be defined, but only one can be active at a time. You need to stop one before you can start the other.
- Neither VRFs, management ports, nor private VLANs can be used as attachment points.
- Only one ACL of each type (IPv4, IPv6, MAC) is allowed in a Wireshark class map. There can be a maximum of three ACLs in a class map: one for IPv4, one for IPv6, and the other for MAC.
- Wireshark cannot capture packets on a destination SPAN port.
- Wireshark will stop capturing when one of the attachment points (interfaces) attached to a capture point stops working. For example, if the device that is associated with an attachment point is unplugged from the switch. To resume capturing, the capture must be restarted manually.
- CPU-injected packets are considered control plane packets. Therefore, these types of packets will not be captured on an interface egress capture.
- MAC ACL is only used for non-IP packets such as ARP. It will not be supported on a Layer 3 port or SVI.
- IPv6-based ACLs are not supported in VACL.
- Layer 2 and Layer 3 EtherChannels are not supported.
- ACL logging and Wireshark are incompatible. Once Wireshark is activated, it takes priority. All traffic, including that being captured by ACL logging on any ports, will be redirected to Wireshark. We

recommended that you deactivate ACL logging before starting Wireshark. Otherwise, Wireshark traffic will be contaminated by ACL logging traffic.

- Wireshark does not capture packets dropped by floodblock.
- If you capture both PACL and RACL on the same port, only one copy is sent to the CPU. If you capture a DTLS-encrypted CAPWAP interface, two copies are sent to Wireshark, one encrypted and the other decrypted. The same behavior will occur if we capture a Layer 2 interface carrying DTLS-encrypted CAPWAP traffic. The core filter is based on the outer CAPWAP header.

Information About Wireshark

Wireshark Overview

Wireshark is a packet analyzer program, formerly known as Ethereal, that supports multiple protocols and presents information in a text-based user interface.

The ability to capture and analyze traffic provides data on network activity. Prior to Cisco IOS Release XE 3.3.0(SE), only two features addressed this need: SPAN and debug platform packet. Both have limitations. SPAN is ideal for capturing packets, but can only deliver them by forwarding them to some specified local or remote destination; it provides no local display or analysis support. The **debug platform packet** command is specific to the Catalyst 4500 series and only works on packets that come from the software process-forwarding path. Also, the **debug platform packet** command has limited local display capabilities and no analysis support.

So the need exists for a traffic capture and analysis mechanism that is applicable to both hardware and software forwarded traffic and that provides strong packet capture, display, and analysis support, preferably using a well known interface.

Wireshark dumps packets to a file using a well known format called .pcap, and is applied or enabled on individual interfaces. You specify an interface in EXEC mode along with the filter and other parameters. The Wireshark application is applied only when you enter a **start** command, and is removed only when Wireshark stops capturing packets either automatically or manually.

Capture Points

A capture point is the central policy definition of the Wireshark feature. The capture point describes all of the characteristics associated with a given instance of Wireshark: which packets to capture, where to capture them from, what to do with the captured packets, and when to stop. Capture points can be modified after creation, and do not become active until explicitly activated with a **start** command. This process is termed activating the capture point or starting the capture point. Capture points are identified by name and can also be manually or automatically deactivated or stopped.

Multiple capture points can be defined, but only one can be active at a time. You need to stop one before you can start the other.

Attachment Points

An attachment point is a point in the logical packet process path associated with a capture point. An attachment point is an attribute of the capture point. Packets that impact an attachment point are tested against capture

point filters; packets that match are copied and sent to the associated Wireshark instance of the capture point. A specific capture point can be associated with multiple attachment points, with limits on mixing attachment points of different types. Some restrictions apply when you specify attachment points of different types. Attachment points are directional (input or output or both) with the exception of the Layer 2 VLAN attachment point, which is always bidirectional.

Filters

Filters are attributes of a capture point that identify and limit the subset of traffic traveling through the attachment point of a capture point, which is copied and passed to Wireshark. To be displayed by Wireshark, a packet must pass through an attachment point, as well as all of the filters associated with the capture point.

A capture point has the following types of filters:

- Core system filter—The core system filter is applied by hardware, and its match criteria is limited by hardware. This filter determines whether hardware-forwarded traffic is copied to software for Wireshark purposes.
- Display filter—The display filter is applied by Wireshark. Packets that fail the display filter are not displayed.

Core System Filter

You can specify core system filter match criteria by using the class map or ACL, or explicitly by using the CLI.

Note When specifying CAPWAP as an attachment point, the core system filter is not used.

In some installations, you need to obtain authorization to modify the switch configuration, which can lead to extended delays if the approval process is lengthy. This can limit the ability of network administrators to monitor and analyze traffic. To address this situation, Wireshark supports explicit specification of core system filter match criteria from the EXEC mode CLI. The disadvantage is that the match criteria that you can specify is a limited subset of what class map supports, such as MAC, IP source and destination addresses, ether-type, IP protocol, and TCP/UDP source and destination ports.

If you prefer to use configuration mode, you can define ACLs or have class maps refer capture points to them. Explicit and ACL-based match criteria are used internally to construct class maps and policy maps.

Note The ACL and class map configuration are part of the system and not aspects of the Wireshark feature.

Display Filter

With the display filter, you can direct Wireshark to further narrow the set of packets to display when decoding and displaying from a .pcap file.

Related Topics

Additional References, on page 161

Actions

Wireshark can be invoked on live traffic or on a previously existing .pcap file. When invoked on live traffic, it can perform four types of actions on packets that pass its display filters:

- Captures to buffer in memory to decode and analyze and store
- Stores to a .pcap file
- Decodes and displays
- Stores and displays

When invoked on a .pcap file only, only the decode and display action is applicable.

Storage of Captured Packets to Buffer in Memory

Packets can be stored in the capture buffer in memory for subsequent decode, analysis, or storage to a .pcap file.

The capture buffer can be in linear or circular mode. In linear mode, new packets are discarded when the buffer is full. In circular mode, if the buffer is full, the oldest packets are discarded to accommodate the new packets. Although the buffer can also be cleared when needed, this mode is mainly used for debugging network traffic.

Note If you have more than one capture that is storing packets in a buffer, clear the buffer before starting a new capture to avoid memory loss.

Storage of Captured Packets to a .pcap File

Note When WireShark is used on switches in a stack, packet captures can be stored only on flash or USB flash devices connected to the active switch.

For example, if flash1 is connected to the active switch, and flash2 is connected to the secondary switch, only flash1 can be used to store packet captures.

Attempts to store packet captures on devices other than flash or USB flash devices connected to the active switch will probably result in errors.

Wireshark can store captured packets to a .pcap file. The capture file can be located on the following storage devices:

- Switch on-board flash storage (flash:)
- USB drive (usbflash0:)

Note Attempts to store packet captures on unsupported devices or devices not connected to the active switch will probably result in errors.

When configuring a Wireshark capture point, you can associate a filename. When the capture point is activated, Wireshark creates a file with the specified name and writes packets to it. If the file already exists when the file is associated or the capture point is activated, Wireshark queries you as to whether the file can be overwritten. Only one capture point may be associated with a given filename.

If the destination of the Wireshark writing process is full, Wireshark fails with partial data in the file. You must ensure that there is sufficient space in the file system before you start the capture session. With Cisco IOS Release IOS XE 3.3.0(SE), the file system full status is not detected for some storage devices.

You can reduce the required storage space by retaining only a segment, instead of the entire packet. Typically, you do not require details beyond the first 64 or 128 bytes. The default behavior is to store the entire packet.

To avoid possible packet drops when processing and writing to the file system, Wireshark can optionally use a memory buffer to temporarily hold packets as they arrive. Memory buffer size can be specified when the capture point is associated with a .pcap file.

Packet Decoding and Display

Wireshark can decode and display packets to the console. This functionality is possible for capture points applied to live traffic and for capture points applied to a previously existing .pcap file.

Note Decoding and displaying packets may be CPU intensive.

Wireshark can decode and display packet details for a wide variety of packet formats. The details are displayed by entering the **monitor capture name start** command with one of the following keyword options, which place you into a display and decode mode:

- brief—Displays one line per packet (the default).
- detailed—Decodes and displays all the fields of all the packets whose protocols are supported. Detailed modes require more CPU than the other two modes.
- (hexadecimal) dump—Displays one line per packet as a hexadecimal dump of the packet data and the printable characters of each packet.

When you enter the **capture** command with the decode and display option, the Wireshark output is returned to Cisco IOS and displayed on the console unchanged.

Live Traffic Display

Wireshark receives copies of packets from the core system. Wireshark applies its display filters to discard uninteresting packets, and then decodes and displays the remaining packets.

.pcap File Display

Wireshark can decode and display packets from a previously stored .pcap file and direct the display filter to selectively displayed packets.

Packet Storage and Display

Functionally, this mode is a combination of the previous two modes. Wireshark stores packets in the specified .pcap file and decodes and displays them to the console. Only the core filters are applicable here.

Wireshark Capture Point Activation and Deactivation

After a Wireshark capture point has been defined with its attachment points, filters, actions, and other options, it must be activated. Until the capture point is activated, it does not actually capture packets.

Before a capture point is activated, some functional checks are performed. A capture point cannot be activated if it has neither a core system filter nor attachment points defined. Attempting to activate a capture point that does not meet these requirements generates an error.*

Note *When performing a wireless capture with a CAPWAP tunneling interface, the core system filter is not required and cannot be used.

The display filters are specified as needed.

After Wireshark capture points are activated, they can be deactivated in multiple ways. A capture point that is storing only packets to a .pcap file can be halted manually or configured with time or packet limits, after which the capture point halts automatically.

When a Wireshark capture point is activated, a fixed rate policer is applied automatically in the hardware so that the CPU is not flooded with Wireshark-directed packets. The disadvantage of the rate policer is that you cannot capture contiguous packets beyond the established rate even if more resources are available.

Wireshark Features

This section describes how Wireshark features function in the switch environment:

- If port security and Wireshark are applied on an ingress capture, a packet that is dropped by port security will still be captured by Wireshark. If port security is applied on an ingress capture, and Wireshark is applied on an egress capture, a packet that is dropped by port security will not be captured by Wireshark.
- Packets dropped by Dynamic ARP Inspection (DAI) are not captured by Wireshark.
- If a port that is in STP blocked state is used as an attachment point and the core filter is matched, Wireshark will capture the packets that come into the port, even though the packets will be dropped by the switch.
- Classification-based security features—Packets that are dropped by input classification-based security features (such as ACLs and IPSG) are not caught by Wireshark capture points that are connected to attachment points at the same layer. In contrast, packets that are dropped by output classification-based security features are caught by Wireshark capture points that are connected to attachment points at the same layer. The logical model is that the Wireshark attachment point occurs after the security feature lookup on the input side, and symmetrically before the security feature lookup on the output side.

  On ingress, a packet goes through a Layer 2 port, a VLAN, and a Layer 3 port/SVI. On egress, the packet goes through a Layer 3 port/SVI, a VLAN, and a Layer 2 port. If the attachment point is before the point where the packet is dropped, Wireshark will capture the packet. Otherwise, Wireshark will not capture

the packet. For example, Wireshark capture policies connected to Layer 2 attachment points in the input direction capture packets dropped by Layer 3 classification-based security features. Symmetrically, Wireshark capture policies attached to Layer 3 attachment points in the output direction capture packets dropped by Layer 2 classification-based security features.

- Routed ports and switch virtual interfaces (SVIs)—Wireshark cannot capture the output of an SVI because the packets that go out of an SVI's output are generated by CPU. To capture these packets, include the control plane as an attachment point.
- VLANs—When a VLAN is used as a Wireshark attachment point, packets are captured in the input direction only.
- Redirection features—In the input direction, features traffic redirected by Layer 3 (such as PBR and WCCP) are logically later than Layer 3 Wireshark attachment points. Wireshark captures these packets even though they might later be redirected out another Layer 3 interface. Symmetrically, output features redirected by Layer 3 (such as egress WCCP) are logically prior to Layer 3 Wireshark attachment points, and Wireshark will not capture them.
- SPAN—Wireshark and SPAN sources are compatible. You can configure an interface as a SPAN source and as a Wireshark attachment point simultaneously. Configuring a SPAN destination port as a Wireshark attachment point is not supported.
- You can capture packets from a maximum of 1000 VLANs at a time, if no ACLs are applied. If ACLs are applied, the hardware will have less space for Wireshark to use. As a result, the maximum number of VLANs than can be used for packet capture at a time will be lower. Using more than 1000 VLANs tunnels at a time or extensive ACLs might have unpredictable results. For example, mobility may go down.

Note Capturing an excessive number of attachment points at the same time is strongly discouraged because it may cause excessive CPU utilization and unpredictable hardware behavior.

Wireless Packet Capture in Wireshark

- Wireless traffic is encapsulated inside CAPWAP packets. However, capturing only a particular wireless client's traffic inside a CAPWAP tunnel is not supported when using the CAPWAP tunnel as an attachment point. To capture only a particular wireless client's traffic, use the client VLAN as an attachment point and formulate the core filter accordingly.
- Limited decoding of inner wireless traffic is supported. Decoding of inner wireless packets inside encrypted CAPWAP tunnels is not supported.
- No other interface type can be used with the CAPWAP tunneling interface on the same capture point. A CAPWAP tunneling interface and a Level 2 port cannot be attachment points on the same capture point.
- You cannot specify a core filter when capturing packets for Wireshark via the CAPWAP tunnel. However, you can use the Wireshark display filters for filtering wireless client traffic against a specific wireless client.
- You can capture packets from a maximum of 135 CAPWAP tunnels at a time if no ACLs are applied. If ACLs are applied, the hardware memory will have less space for Wireshark to use. As a result, the maximum number of CAPWAP tunnels than can be used for packet capture at a time will be lower.

Using more than 135 CAPWAP tunnels at a time or unsing extensive ACLs might have unpredictable results. For example, mobility may go down.

Note Capturing an excessive number of attachment points at the same time is strongly discouraged because it may cause excessive CPU utilization and unpredictable hardware behavior.

Guidelines for Wireshark

- During Wireshark packet capture, hardware forwarding happens concurrently.
- Before starting a Wireshark capture process, ensure that CPU usage is moderate and that sufficient memory (at least 200 MB) is available.
- If you plan to store packets to a storage file, ensure that sufficient space is available before beginning a Wireshark capture process.
- The CPU usage during Wireshark capture depends on how many packets match the specified conditions and on the intended actions for the matched packets (store, decode and display, or both).
- Where possible, keep the capture to the minimum (limit by packets, duration) to avoid high CPU usage and other undesirable conditions.
- Because packet forwarding typically occurs in hardware, packets are not copied to the CPU for software processing. For Wireshark packet capture, packets are copied and delivered to the CPU, which causes an increase in CPU usage.

  To avoid high CPU usage, do the following:

  - Attach only relevant ports.
  - Use a class map, and secondarily, an access list to express match conditions. If neither is viable, use an explicit, in-line filter.
  - Adhere closely to the filter rules. Restrict the traffic type (such as, IPv4 only) with a restrictive, rather than relaxed ACL, which elicits unwanted traffic.

- Always limit packet capture to either a shorter duration or a smaller packet number. The parameters of the capture command enable you to specify the following:

  - Capture duration
  - Number of packets captured
  - File size
  - Packet segment size

- Run a capture session without limits if you know that very little traffic matches the core filter.
- You might experience high CPU (or memory) usage if:

  - You leave a capture session enabled and unattended for a long period of time, resulting in unanticipated bursts of traffic.

- You launch a capture session with ring files or capture buffer and leave it unattended for a long time, resulting in performance or system health issues.

- During a capture session, watch for high CPU usage and memory consumption due to Wireshark that may impact switch performance or health. If these situations arise, stop the Wireshark session immediately.
- Avoid decoding and displaying packets from a .pcap file for a large file. Instead, transfer the .pcap file to a PC and run Wireshark on the PC.
- You can define up to eight Wireshark instances. An active **show** command that decodes and displays packets from a .pcap file or capture buffer counts as one instance. However, only one of the instances can be active.
- Whenever an ACL that is associated with a running capture is modified, you must restart the capture for the ACL modifications to take effect. If you do not restart the capture, it will continue to use the original ACL as if it had not been modified.
- To avoid packet loss, consider the following:
  - Use store-only (when you do not specify the display option) while capturing live packets rather than decode and display, which is an CPU-intensive operation (especially in detailed mode).
  - If you have more than one capture that is storing packets in a buffer, clear the buffer before starting a new capture to avoid memory loss.
  - If you use the default buffer size and see that you are losing packets, you can increase the buffer size to avoid losing packets.
  - Writing to flash disk is a CPU-intensive operation, so if the capture rate is insufficient, you may want to use a buffer capture.
  - The Wireshark capture session operates normally in streaming mode where packets are both captured and processed. However, when you specify a buffer size of at least 32 MB, the session automatically turns on lock-step mode in which a Wireshark capture session is split into two phases: capture and process. In the capture phase, the packets are stored in the temporary buffer. The duration parameter in lock-step mode serves as capture duration rather than session duration. When the buffer is full or the capture duration or packet limit has been attained, a session transitions to the process phase, wherein it stops accepting packets and starts processing packets in the buffer. You can also stop the capture manually. You will see a message in the output when the capture stops. With this second approach (lock-step mode), a higher capture throughput can be achieved.

    Note If you are capturing packets to a buffer, there is no file storage defined. Hence, you must export your capture from the buffer to a static storage file. Use the **monitor capture** *capture-name* **export** *file-location* : *file-name* command.

  - The streaming capture mode supports approximately 1000 pps; lock-step mode supports approximately 2 Mbps (measured with 256-byte packets). When the matching traffic rate exceeds this number, you may experience packet loss.

- If you want to decode and display live packets in the console window, ensure that the Wireshark session is bounded by a short capture duration.

Note **Warning**: A Wireshark session with either a longer duration limit or no capture duration (using a terminal with no auto-more support using the **term len 0** command) may make the console or terminal unusable.

- When using Wireshark to capture live traffic that leads to high CPU, usage, consider applying a QoS policy temporarily to limit the actual traffic until the capture process concludes.
- All Wireshark-related commands are in EXEC mode; no configuration commands exist for Wireshark.

  If you need to use access list or class-map in the Wireshark CLI, you must define an access list and class map with configuration commands.
- No specific order applies when defining a capture point; you can define capture point parameters in any order, provided that CLI allows this. The Wireshark CLI allows as many parameters as possible on a single line. This limits the number of commands required to define a capture point.
- All parameters except attachment points take a single value. Generally, you can replace the value with a new one by reentering the command. After user confirmation, the system accepts the new value and overrides the older one. A **no** form of the command is unnecessary to provide a new value, but it is necessary to remove a parameter.
- Wireshark allows you to specify one or more attachment points. To add more than one attachment point, reenter the command with the new attachment point. To remove an attachment point, use the **no** form of the command. You can specify an interface range as an attachment point. For example, enter **monitor capture mycap interface GigabitEthernet1/0/1 in** where interface GigabitEthernet1/0/1 is an attachment point.

  If you also need to attach interface GigabitEthernet1/0/2, specify it in another line as follows:

  monitor capture mycap interface GigabitEthernet1/0/2 in
- You can modify any of the parameters of a capture point while a session is active, but you must restart the session for the modifications to take effect.
- The action you want to perform determines which parameters are mandatory. The Wireshark CLI allows you to specify or modify any parameter prior to entering the **start** command. When you enter the **start** command, Wireshark will start only after determining that all mandatory parameters have been provided.
- If the capture file already exists, it provides a warning and receives confirmation before proceeding. This prevents you from mistakenly overwriting a file.
- The core filter can be an explicit filter, access list, or class map. Specifying a newer filter of these types replaces the existing one.

  Note A core filter is required except when using a CAPWAP tunnel interface as a capture point attachment point.
- You can terminate a Wireshark session with an explicit **stop** command or by entering **q** in automore mode. The session could terminate itself automatically when a stop condition such as duration or packet capture limit is met.

Default Wireshark Configuration

The table below shows the default Wireshark configuration.

Feature	Default Setting
Duration	No limit
Packets	No limit
Packet-length	No limit (full packet)
File size	No limit
Ring file storage	No
Buffer storage mode	Linear

How to Configure Wireshark

To configure Wireshark, perform these basic steps.

1. Define a capture point.
2. (Optional) Add or modify the capture point's parameters.
3. Activate or deactivate a capture point.
4. Delete the capture point when you are no longer using it.

Related Topics

Defining a Capture Point, on page 138
Adding or Modifying Capture Point Parameters, on page 142
Deleting Capture Point Parameters, on page 144
Deleting a Capture Point, on page 145
Activating and Deactivating a Capture Point, on page 146
Clearing the Capture Point Buffer, on page 147

Defining a Capture Point

The example in this procedure defines a very simple capture point. If you choose, you can define a capture point and all of its parameters with one instance of the **monitor capture** command.

Note You must define an attachment point, direction of capture, and core filter to have a functional capture point.

An exception to needing to define a core filter is when you are defining a wireless capture point using a CAPWAP tunneling interface. In this case, you do not define your core filter. It cannot be used.

In privileged EXEC mode, follow these steps to define a capture point.

SUMMARY STEPS

1. **show capwap summary**
2. **monitor capture** {*capture-name*} {**interface** *interface-type interface-id* | **control-plane**} {**in** | **out** | **both**}
3. **monitor capture** {*capture-name*} [**match** {**any** | **ipv4 any any** | **ipv6**} **any any**}]
4. **show monitor capture** {*capture-name*} [**parameter**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show capwap summary** **Example:** `Switch# show capwap summary`	Displays the CAPWAP tunnels available as attachment points for a wireless capture. **Note** Use this command only if you are using a CAPWAP tunnel as an attachment point to perform a wireless capture. See the CAPWAP example in the examples section.
Step 2	**monitor capture** {*capture-name*} {**interface** *interface-type interface-id*	**control-plane**} {**in**	**out**	**both**} **Example:** `Switch# monitor capture mycap interface GigabitEthernet1/0/1 in`	Defines the capture point, specifies the attachment point with which the capture point is associated, and specifies the direction of the capture. The keywords have these meanings: • *capture-name*—Specifies the name of the capture point to be defined (mycap is used in the example). • (Optional) **interface** *interface-type interface-id*—Specifies the attachment point with which the capture point is associated (GigabitEthernet1/0/1 is used in the example). **Note** Optionally, you can define multiple attachment points and all of the parameters for this capture point with this one command instance. These parameters are discussed in the instructions for modifying capture point parameters. Range support is also available both for adding and removing attachment points. Use one of the following for *interface-type*: ◦ **GigabitEthernet**—Specifies the attachment point as GigabitEthernet. ◦ **vlan**—Specifies the attachment point as a VLAN. **Note** Only ingress capture (in) is allowed when using this interface as an attachment point. ◦ **capwap**—Specifies the attachment point as a CAPWAP tunnel.

	Command or Action	Purpose
		Note When using this interface as an attachment point, a core filter cannot be used. • (Optional) **control-plane—**Specifies the control plane as an attachment point. • **in** | **out** | **both—**Specifies the direction of capture.
Step 3	**monitor capture** {*capture-name*}[**match** {**any** | **ipv4 any any** | **ipv6**} **any any**}] **Example:** `Switch# monitor capture mycap interface GigabitEthernet1/0/1 in match any`	Defines the core system filter. **Note** When using the CAPWAP tunneling interface as an attachment point, do not perform this step because a core filter cannot be used. The keywords have these meanings: • *capture-name*—Specifies the name of the capture point to be defined (mycap is used in the example). • **match—**Specifies a filter. The first filter defined is the core filter. **Note** A capture point cannot be activated if it has neither a core system filter nor attachment points defined. Attempting to activate a capture point that does not meet these requirements generates an error. • **ipv4—**Specifies an IP version 4 filter. • **ipv6—**Specifies an IP version 6 filter.

Step 4	**show monitor capture** {*capture-name*}[**parameter**]
	
	Example:
	`Switch# show monitor capture mycap parameter`
	`monitor capture mycap interface GigabitEthernet1/0/1 in`
	`monitor capture mycap match any`
		Displays the capture point parameters that you defined in Step 1 and confirms that you defined a capture point.

To define a capture point with a CAPWAP attachment point:

```
Switch# show capwap summary

CAPWAP Tunnels General Statistics:
  Number of Capwap Data Tunnels       = 1
  Number of Capwap Mobility Tunnels   = 0
  Number of Capwap Multicast Tunnels  = 0


Name   APName                           Type PhyPortIf Mode      McastIf
------ -------------------------------- ---- --------- --------- -------
Ca0    AP442b.03a9.6715                 data Gi3/0/6   unicast   -


Name   SrcIP           SrcPort DestIP          DstPort DtlsEn MTU   Xact
------ --------------- ------- --------------- ------- ------ ----- ----
Ca0    10.10.14.32     5247    10.10.14.2      38514   No     1449  0

Switch# monitor capture mycap interface capwap 0 both
```

```
Switch# monitor capture mycap file location flash:mycap.pcap
Switch# monitor capture mycap file buffer-size 1
Switch# monitor capture mycap start

*Aug 20 11:02:21.983: %BUFCAP-6-ENABLE: Capture Point mycap enabled.on

Switch# show monitor capture mycap parameter
   monitor capture mycap interface capwap  0 in
   monitor capture mycap interface capwap  0 out
   monitor capture mycap file location flash:mycap.pcap buffer-size 1
Switch#
Switch# show monitor capture mycap

Status Information for Capture mycap
  Target Type:
  Interface: CAPWAP,
    Ingress:
 0
    Egress:
 0
   Status : Active
  Filter Details:
    Capture all packets
  Buffer Details:
   Buffer Type: LINEAR (default)
  File Details:
   Associated file name: flash:mycap.pcap
   Size of buffer(in MB): 1
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Packets per second: 0 (no limit)
   Packet sampling rate: 0 (no sampling)
Switch#
Switch# show monitor capture file flash:mycap.pcap
  1   0.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
  2   0.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
  3   2.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
  4   2.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
  5   3.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
  6   4.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
  7   4.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
  8   5.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
  9   5.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
 10   6.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
 11   8.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
 12   9.225986   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 13   9.225986   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 14   9.225986   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 15   9.231998   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 16   9.231998   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 17   9.231998   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 18   9.236987   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 19  10.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
 20  10.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
 21  12.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
 Flags=........
 22  12.239993   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 23  12.244997   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
```

```
24  12.244997   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
25  12.250994   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
26  12.256990   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
27  12.262987   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
28  12.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
Flags=........
29  12.802012   10.10.14.3 -> 10.10.14.255 NBNS Name query NB WPAD.<00>
30  13.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0, FN=0,
Flags=........
```

What to Do Next

You can add additional attachment points, modify the parameters of your capture point, then activate it, or if you want to use your capture point just as it is, you can now activate it.

Note You cannot change a capture point's parameters using the methods presented in this topic.

Related Topics

How to Configure Wireshark, on page 138
Adding or Modifying Capture Point Parameters, on page 142
Deleting Capture Point Parameters, on page 144
Deleting a Capture Point, on page 145
Activating and Deactivating a Capture Point, on page 146
Adding or Modifying Capture Point Parameters, on page 142

Adding or Modifying Capture Point Parameters

Although listed in sequence, the steps to specify values for the parameters can be executed in any order. You can also specify them in one, two, or several lines. Except for attachment points, which can be multiple, you can replace any value with a more recent value by redefining the same option.

In privileged EXEC mode, follow these steps to modify a capture point's parameters.

Before You Begin

A capture point must be defined before you can use these instructions.

SUMMARY STEPS

1. **monitor capture** {*capture-name*} **match** {**any** | **mac** *mac-match-string* | **ipv4** {**any** | **host** | **protocol**} {**any** | **host**} | **ipv6** {**any** | **host** | **protocol**} {**any** | **host**}}
2. **monitor capture** {*capture-name*} **limit** {[**duration** *seconds*][**packet-length** *size*][**packets** *num*]}
3. **monitor capture** {*capture-name*} **file** {**location** *filename*}
4. **monitor capture** {*capture-name*} **file** {**buffer-size** *size*}
5. **show monitor capture** {*capture-name*}[**parameter**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**monitor capture** {*capture-name*} **match** {**any** | **mac** *mac-match-string* | **ipv4** {**any** | **host** | **protocol**}{**any** | **host**} | **ipv6** {**any** | **host** | **protocol**}{**any** | **host**}} **Example:** `Switch# monitor capture mycap match ipv4 any any`	Defines the core system filter (**ipv4 any any**), defined either explicitly, through ACL or through a class map. **Note** If you are defining a wireless capture point using a CAPWAP tunneling interface, this command will have no effect, so it should not be used.
Step 2	**monitor capture** {*capture-name*} **limit** {[**duration** *seconds*][**packet-length** *size*][**packets** *num*]} **Example:** `Switch# monitor capture mycap limit duration 60 packet-len 400`	Specifies the session limit in seconds (60), packets captured, or the packet segment length to be retained by Wireshark (400).
Step 3	**monitor capture** {*capture-name*} **file** {**location** *filename*} **Example:** `Switch# monitor capture mycap file location flash:mycap.pcap`	Specifies the file association, if the capture point intends to capture packets rather than only display them.
Step 4	**monitor capture** {*capture-name*} **file** {**buffer-size** *size*} **Example:** `Switch# monitor capture mycap file buffer-size 100`	Specifies the size of the memory buffer used by Wireshark to handle traffic bursts.

Step 5	**show monitor capture** {*capture-name*}[**parameter**]
	
	Example:
	`Switch# show monitor capture mycap parameter`
	`monitor capture mycap interface GigabitEthernet1/0/1 in`
	`monitor capture mycap match ipv4 any any`
	`monitor capture mycap limit duration 60 packet-len 400`
	`monitor capture point mycap file location bootdisk:mycap.pcap`
	`monitor capture mycap file buffer-size 100`
		Displays the capture point parameters that you defined previously.

Examples

Modifying Parameters

Associating or Disassociating a Capture File

```
Switch# monitor capture point mycap file location flash:mycap.pcap
Switch# no monitor capture mycap file
```

Specifying a Memory Buffer Size for Packet Burst Handling

```
Switch# monitor capture mycap buffer size 100
```

Defining an Explicit Core System Filter to Match Both IPv4 and IPv6

```
Switch# monitor capture mycap match any
```

What to Do Next

if your capture point contains all of the parameters you want, activate it.

Related Topics

How to Configure Wireshark, on page 138
Defining a Capture Point, on page 138
Defining a Capture Point, on page 138
Deleting Capture Point Parameters, on page 144
Deleting a Capture Point, on page 145

Deleting Capture Point Parameters

Although listed in sequence, the steps to delete parameters can be executed in any order. You can also delete them in one, two, or several lines. Except for attachment points, which can be multiple, you can delete any parameter.

In privileged EXEC mode, follow these steps to delete a capture point's parameters.

Before You Begin

A capture point parameter must be defined before you can use these instructions to delete it.

SUMMARY STEPS

1. **no monitor capture** {*capture-name*} **match**
2. **no monitor capture** {*capture-name*} **limit** [**duration**][**packet-length**][**packets**]
3. **no monitor capture** {*capture-name*} **file** [**location**] [**buffer-size**]
4. **show monitor capture** {*capture-name*}[**parameter**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**no monitor capture** {*capture-name*} **match** **Example:** `Switch# no monitor capture mycap match`	Deletes all filters defined on capture point (mycap).

Step 2	**no monitor capture** {*capture-name*} **limit** [**duration**][**packet-length**][**packets**]
	
	Example:
	`Switch# no monitor capture mycap limit duration packet-len`
	`Switch# no monitor capture mycap limit`
		Deletes the session time limit and the packet segment length to be retained by Wireshark. It leaves other specified limits in place.
		
		Deletes all limits on Wireshark.

Step 3	**no monitor capture** {*capture-name*} **file** [**location**] [**buffer-size**]	Deletes the file association. The capture point will no longer capture packets. It will only display them.

	Command or Action	Purpose

	Example:
	`Switch# no monitor capture mycap file`
	`Switch# no monitor capture mycap file location`
		Deletes the file location association. The file location will no longer be associated with the capture point. However, other defined fille association will be unaffected by this action.

Step 4	**show monitor capture** {*capture-name*}[**parameter**]
	
	Example:
	`Switch# show monitor capture mycap parameter`
	`monitor capture mycap interface GigabitEthernet1/0/1 in`
		Displays the capture point parameters that remain defined after your parameter deletion operations. This command can be run at any point in the procedure to see what parameters are associated with a capture point.

What to Do Next

If your capture point contains all of the parameters you want, activate it.

Related Topics

How to Configure Wireshark, on page 138
Defining a Capture Point, on page 138
Adding or Modifying Capture Point Parameters, on page 142

Deleting a Capture Point

In privileged EXEC mode, follow these steps to delete a capture point.

Before You Begin

A capture point must be defined before you can use these instructions to delete it.

SUMMARY STEPS

1. **no monitor capture** {*capture-name*}
2. **show monitor capture** {*capture-name*}[**parameter**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**no monitor capture** {*capture-name*} **Example:** `Switch# no monitor capture mycap`	Deletes the specified capture point (mycap).

	Command or Action	Purpose

Step 2	**show monitor capture** {*capture-name*}[**parameter**]
	
	Example:
	`Switch# show monitor capture mycap parameter`
	`Capture mycap does not exist`
		Displays a message indicating that the specified capture point does not exist because it has been deleted.

What to Do Next

You can define a new capture point with the same name as the one you deleted. These instructions are usually performed when one wants to start over with defining a capture point.

Related Topics

How to Configure Wireshark, on page 138
Defining a Capture Point, on page 138
Adding or Modifying Capture Point Parameters, on page 142

Activating and Deactivating a Capture Point

In privileged EXEC mode, follow these steps to activate or deactivate a capture point.

Before You Begin

A capture point cannot be activated unless an attachment point and a core system filter have been defined and the associated filename (if any) does not already exist. A capture point with no associated filename can only be activated to display. If no capture or display filters are specified, all of the packets captured by the core system filter are displayed. The default display mode is brief.

Note When using a CAPWAP tunneling interface as an attachment point, core filters are not used, so there is no requirement to define them in this case.

SUMMARY STEPS

1. **monitor capture** {*capture-name*} **start**[**display** [**display-filter** *filter-string*]][**brief** | **detailed** | **dump**]
2. **monitor capture** {*capture-name*} **stop**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**monitor capture** {*capture-name*} **start**[**display** [**display-filter** *filter-string*]][**brief** | **detailed** | **dump**] **Example:** `Switch# monitor capture mycap start display display-filter "stp"`	Activates a capture point and filters the display, so only packets containing "stp" are displayed.
Step 2	**monitor capture** {*capture-name*} **stop** **Example:** `Switch# monitor capture name stop`	Deactivates a capture point.

Related Topics

How to Configure Wireshark, on page 138
Defining a Capture Point, on page 138

Clearing the Capture Point Buffer

In privileged EXEC mode, follow these steps to clear the buffer contents or save them to an external file for storage.

Note If you have more than one capture that is storing packets in a buffer, clear the buffer before starting a new capture to avoid memory loss.

SUMMARY STEPS

1. **monitor capture** {*capture-name*} [**clear** | **export** *filename*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**monitor capture** {*capture-name*} [**clear** | **export** *filename*] **Example:** `Switch# monitor capture mycap clear`	Clears capture buffer contents or stores the packets to a file.

Examples: Capture Point Buffer Handling

Exporting Capture to a File

```
Switch# monitor capture mycap export flash:mycap.pcap

Storage configured as File for this capture
```

Clearing Capture Point Buffer

```
Switch# monitor capture mycap clear

 Capture configured with file options
```

Related Topics

How to Configure Wireshark, on page 138

Monitoring Wireshark

The commands in this table are used to monitor Wireshark.

Command	Purpose
show monitor capture [*capture-name*]	Displays the capture point state so that you can see what capture points are defined, what their attributes are, and whether they are active. When capture point name is specified, it displays specific capture point's details.
show monitor capture [*capture-name* **parameter**]	Displays the capture point parameters.
show capwap summary	Displays all the CAPWAP tunnels on the switch. Use this command to determine which CAPWAP tunnels are available to use for a wireless capture.

Configuration Examples for Wireshark

Example: Displaying a Brief Output from a .pcap File

You can display the output from a .pcap file by entering:

```
Switch# show monitor capture file flash:mycap.pcap
  1   0.000000   10.1.1.140 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2   1.000000   10.1.1.141 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3   2.000000   10.1.1.142 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4   3.000000   10.1.1.143 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5   4.000000   10.1.1.144 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6   5.000000   10.1.1.145 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

```
  7   6.000000   10.1.1.146 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8   7.000000   10.1.1.147 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9   8.000000   10.1.1.148 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 10   9.000000   10.1.1.149 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 11  10.000000   10.1.1.150 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 12  11.000000   10.1.1.151 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 13  12.000000   10.1.1.152 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 14  13.000000   10.1.1.153 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 15  14.000000   10.1.1.154 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 16  15.000000   10.1.1.155 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 17  16.000000   10.1.1.156 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 18  17.000000   10.1.1.157 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 19  18.000000   10.1.1.158 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 20  19.000000   10.1.1.159 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 21  20.000000   10.1.1.160 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 22  21.000000   10.1.1.161 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 23  22.000000   10.1.1.162 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 24  23.000000   10.1.1.163 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 25  24.000000   10.1.1.164 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 26  25.000000   10.1.1.165 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 27  26.000000   10.1.1.166 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 28  27.000000   10.1.1.167 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 29  28.000000   10.1.1.168 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 30  29.000000   10.1.1.169 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 31  30.000000   10.1.1.170 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 32  31.000000   10.1.1.171 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 33  32.000000   10.1.1.172 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 34  33.000000   10.1.1.173 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 35  34.000000   10.1.1.174 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 36  35.000000   10.1.1.175 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 37  36.000000   10.1.1.176 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 38  37.000000   10.1.1.177 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 39  38.000000   10.1.1.178 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 40  39.000000   10.1.1.179 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 41  40.000000   10.1.1.180 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 42  41.000000   10.1.1.181 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

```
 43  42.000000   10.1.1.182 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 44  43.000000   10.1.1.183 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 45  44.000000   10.1.1.184 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 46  45.000000   10.1.1.185 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 47  46.000000   10.1.1.186 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 48  47.000000   10.1.1.187 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 49  48.000000   10.1.1.188 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 50  49.000000   10.1.1.189 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 51  50.000000   10.1.1.190 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 52  51.000000   10.1.1.191 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 53  52.000000   10.1.1.192 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 54  53.000000   10.1.1.193 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 55  54.000000   10.1.1.194 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 56  55.000000   10.1.1.195 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 57  56.000000   10.1.1.196 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 58  57.000000   10.1.1.197 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 59  58.000000   10.1.1.198 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

Example: Displaying Detailed Output from a .pcap File

You can display the detailed .pcap file output by entering:

```
Switch# show monitor capture file flash:mycap.pcap detailed

Frame 1: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
    Arrival Time: Mar 21, 2012 14:35:09.111993000 PDT
    Epoch Time: 1332365709.111993000 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: 256 bytes (2048 bits)
    Capture Length: 256 bytes (2048 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:data]
Ethernet II, Src: 00:00:00:00:03:01 (00:00:00:00:03:01), Dst: 54:75:d0:3a:85:3f
(54:75:d0:3a:85:3f)
    Destination: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
        Address: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Source: 00:00:00:00:03:01 (00:00:00:00:03:01)
        Address: 00:00:00:00:03:01 (00:00:00:00:03:01)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Type: IP (0x0800)
    Frame check sequence: 0x03b07f42 [incorrect, should be 0x08fcee78]
Internet Protocol, Src: 10.1.1.140 (10.1.1.140), Dst: 20.1.1.2 (20.1.1.2)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
```

```
        .... ...0 = ECN-CE: 0
    Total Length: 238
    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: 64
    Protocol: UDP (17)
    Header checksum: 0x5970 [correct]
        [Good: True]
        [Bad: False]
    Source: 10.1.1.140 (10.1.1.140)
    Destination: 20.1.1.2 (20.1.1.2)
User Datagram Protocol, Src Port: 20001 (20001), Dst Port: 20002 (20002)
    Source port: 20001 (20001)
    Destination port: 20002 (20002)
    Length: 218
    Checksum: 0x6e2b [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
Data (210 bytes)

0000  00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f   ................
0010  10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f   ................
0020  20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f    !"#$%&'()*+,-./
0030  30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f   0123456789:;<=>?
0040  40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f   @ABCDEFGHIJKLMNO
0050  50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f   PQRSTUVWXYZ[\]^_
0060  60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f   `abcdefghijklmno
0070  70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f   pqrstuvwxyz{|}~.
0080  80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f   ................
0090  90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f   ................
00a0  a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af   ................
00b0  b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf   ................
00c0  c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf   ................
00d0  d0 d1                                            ..
    Data: 000102030405060708090a0b0c0d0e0f1011121314151617...
    [Length: 210]

Frame 2: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
    Arrival Time: Mar 21, 2012 14:35:10.111993000 PDT
Example: Displaying a Hexadecimal Dump Output from a .pcap File
You can display the hexadecimal dump output by entering:
Switch# show monitor capture file bootflash:mycap.pcap dump
  1   0.000000   10.1.1.140 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 70 0a 01 01 8c 14 01   ......@.Yp......
0020  01 02 4e 21 4e 22 00 da 6e 2b 00 01 02 03 04 05   ..N!N"..n+......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 03 b0 7f 42   ...............B

  2   1.000000   10.1.1.141 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 6f 0a 01 01 8d 14 01   ......@.Yo......
0020  01 02 4e 21 4e 22 00 da 6e 2a 00 01 02 03 04 05   ..N!N"..n*......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
```

```
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 95 2c c3 3f   .............,.?

  3   2.000000   10.1.1.142 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 6e 0a 01 01 8e 14 01   ......@.Yn......
0020  01 02 4e 21 4e 22 00 da 6e 29 00 01 02 03 04 05   ..N!N"..n)......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 6c f8 dc 14   ............l...

  4   3.000000   10.1.1.143 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 6d 0a 01 01 8f 14 01   ......@.Ym......
0020  01 02 4e 21 4e 22 00 da 6e 28 00 01 02 03 04 05   ..N!N"..n(......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
Example: Displaying Packets from a .pcap File with a Display Filter
You can display the .pcap file packets output by entering:
Switch# show monitor capture file bootflash:mycap.pcap display-filter "ip.src == 10.1.1.140"
 dump
  1   0.000000   10.1.1.140 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 70 0a 01 01 8c 14 01   ......@.Yp......
0020  01 02 4e 21 4e 22 00 da 6e 2b 00 01 02 03 04 05   ..N!N"..n+......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 03 b0 7f 42   ...............B
```

Example: Simple Capture and Display

This example shows how to monitor traffic in the Layer 3 interface Gigabit Ethernet 1/0/1:

Step 1: Define a capture point to match on the relevant traffic by entering:

```
Switch# monitor capture mycap interface GigabitEthernet1/0/1 in
Switch# monitor capture mycap match ipv4 any any
```

```
Switch# monitor capture mycap limit duration 60 packets 100
Switch# monitor capture mycap buffer size 100
```

To avoid high CPU utilization, a low packet count and duration as limits has been set.

Step 2: Confirm that the capture point has been correctly defined by entering:

```
Switch# show monitor capture mycap parameter
   monitor capture mycap interface GigabitEthernet1/0/1 in
   monitor capture mycap match ipv4  any any
   monitor capture mycap buffer size 100
   monitor capture mycap limit packets 100 duration 60

Switch# show monitor capture mycap

Status Information for Capture mycap
  Target Type:
   Interface: GigabitEthernet1/0/1, Direction: in
   Status : Inactive
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  Buffer Details:
   Buffer Type: LINEAR (default)
   Buffer Size (in MB): 100
  File Details:
   File not associated
  Limit Details:
   Number of Packets to capture: 100
   Packet Capture duration: 60
   Packet Size to capture: 0 (no limit)
   Packets per second: 0 (no limit)
   Packet sampling rate: 0 (no sampling)
```

Step 3: Start the capture process and display the results.

```
Switch# monitor capture mycap start display
  0.000000    10.1.1.30 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000    10.1.1.31 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000    10.1.1.32 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000    10.1.1.33 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000    10.1.1.34 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000    10.1.1.35 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000    10.1.1.36 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000    10.1.1.37 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.38 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.39 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

Step 4: Delete the capture point by entering:

```
Switch# no monitor capture mycap
```

Example: Simple Capture and Store

This example shows how to capture packets to a filter:

Step 1: Define a capture point to match on the relevant traffic and associate it to a file by entering:

```
Switch# monitor capture mycap interface GigabitEthernet1/0/1 in
Switch# monitor capture mycap match ipv4 any any
Switch# monitor capture mycap limit duration 60 packets 100
Switch# monitor capture mycap file location flash:mycap.pcap
```

Step 2: Confirm that the capture point has been correctly defined by entering:

```
Switch# show monitor capture mycap parameter
   monitor capture mycap interface GigabitEthernet1/0/1 in
   monitor capture mycap match ipv4  any any
```

```
   monitor capture mycap file location flash:mycap.pcap
   monitor capture mycap limit packets 100 duration 60

Switch# show monitor capture mycap

Status Information for Capture mycap
  Target Type:
   Interface: GigabitEthernet1/0/1, Direction: in
   Status : Inactive
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  Buffer Details:
   Buffer Type: LINEAR (default)
  File Details:
   Associated file name: flash:mycap.pcap
  Limit Details:
   Number of Packets to capture: 100
   Packet Capture duration: 60
   Packet Size to capture: 0 (no limit)
   Packets per second: 0 (no limit)
   Packet sampling rate: 0 (no sampling)
```

Step 3: Launch packet capture by entering:

```
Switch# monitor capture mycap start
```

Step 4: After sufficient time has passed, stop the capture by entering:

```
Switch# monitor capture mycap stop
```

Note Alternatively, you could allow the capture operation stop automatically after the time has elapsed or the packet count has been met.

The mycap.pcap file now contains the captured packets.

Step 5: Display the packets by entering:

```
Switch# show monitor capture file flash:mycap.pcap

  0.000000    10.1.1.30 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000    10.1.1.31 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000    10.1.1.32 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000    10.1.1.33 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000    10.1.1.34 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000    10.1.1.35 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000    10.1.1.36 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000    10.1.1.37 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.38 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.39 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

Step 6: Delete the capture point by entering:

```
Switch# no monitor capture mycap
```

Example: Using Buffer Capture

This example shows how to use buffer capture:

Step 1: Launch a capture session with the buffer capture option by entering:

```
Switch# monitor capture mycap interface GigabitEthernet1/0/1 in
Switch# monitor capture mycap match ipv4 any any
```

```
Switch# monitor capture mycap buffer circular size 1
Switch# monitor capture mycap start
```

Step 2: Determine whether the capture is active by entering:

```
Switch# show monitor capture mycap

Status Information for Capture mycap
  Target Type:
   Interface: GigabitEthernet1/0/1, Direction: in
   Status : Active
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  Buffer Details:
   Buffer Type: CIRCULAR
   Buffer Size (in MB): 1
  File Details:
   File not associated
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Packets per second: 0 (no limit)
   Packet sampling rate: 0 (no sampling)
```

Step 3: Display the packets in the buffer by entering:

```
Switch# show monitor capture mycap buffer brief

  0.000000   10.1.1.215 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000   10.1.1.216 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000   10.1.1.217 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000   10.1.1.218 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000   10.1.1.219 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000   10.1.1.220 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000   10.1.1.221 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000   10.1.1.222 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000   10.1.1.223 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000   10.1.1.224 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 10.000000   10.1.1.225 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 11.000000   10.1.1.226 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 12.000000   10.1.1.227 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 13.000000   10.1.1.228 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 14.000000   10.1.1.229 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 15.000000   10.1.1.230 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 16.000000   10.1.1.231 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 17.000000   10.1.1.232 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 18.000000   10.1.1.233 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 19.000000   10.1.1.234 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 20.000000   10.1.1.235 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 21.000000   10.1.1.236 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

Notice that the packets have been buffered.

Step 4: Display the packets in other display modes.

```
Switch# show monitor capture mycap buffer detailed

Frame 1: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
    Arrival Time: Apr 15, 2012 15:50:02.398966000 PDT
    Epoch Time: 1334530202.398966000 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: 256 bytes (2048 bits)
    Capture Length: 256 bytes (2048 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:data]
Ethernet II, Src: 00:00:00:00:03:01 (00:00:00:00:03:01), Dst: 54:75:d0:3a:85:3f
```

```
(54:75:d0:3a:85:3f)
    Destination: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
        Address: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Source: 00:00:00:00:03:01 (00:00:00:00:03:01)
        Address: 00:00:00:00:03:01 (00:00:00:00:03:01)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
...
Switch# show monitor capture mycap buffer dump

  0.000000   10.1.1.215 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 25 0a 01 01 d7 14 01   ......@.Y%......
0020  01 02 4e 21 4e 22 00 da 6d e0 00 01 02 03 04 05   ..N!N"..m.......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 03 3e d0 33   ..............>.3
```

Step 5a: Clear the buffer by entering:

```
Switch# monitor capture mycap clear
```

Step 5b: Wait for 10 seconds.

Step 5c: Stop the traffic by entering:

```
Switch# monitor capture mycap stop
```

Step 6: Confirm that the same set of packets are displayed after this time gap by entering:

```
Switch# show monitor capture mycap buffer brief

  0.000000     10.1.1.2 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000     10.1.1.3 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000     10.1.1.4 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000     10.1.1.5 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000     10.1.1.6 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000     10.1.1.7 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000     10.1.1.8 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000     10.1.1.9 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.10 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.11 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

Step 7: Wait for 10 seconds, then confirm that the same set of packets are displayed after this time gap by entering:

```
Switch# show monitor capture mycap buffer brief

  0.000000     10.1.1.2 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000     10.1.1.3 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000     10.1.1.4 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000     10.1.1.5 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000     10.1.1.6 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000     10.1.1.7 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000     10.1.1.8 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000     10.1.1.9 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.10 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.11 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

Step 8: Repeat Step 7.

Step 9: Clear the buffer by entering:

```
Switch# monitor capture mycap clear
```

Step 10: Confirm that the buffer is now empty by entering:

```
Switch# show monitor capture mycap buffer brief
```

Step 11: Wait about 10 seconds, then display the buffer contents by entering:

```
Switch# show monitor capture mycap buffer brief
```

Step 12: Restart the traffic, wait for 10 seconds, then display the buffer contents by entering:

```
Switch# monitor capture mycap start
wait for 10 seconds...
Switch# show monitor capture mycap buffer brief

  0.000000     10.1.1.2 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000     10.1.1.3 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000     10.1.1.4 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000     10.1.1.5 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000     10.1.1.6 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000     10.1.1.7 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000     10.1.1.8 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000     10.1.1.9 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.10 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.11 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

Step 13: Store the buffer contents to the mycap1.pcap file in the internal flash: storage device by entering:

```
Switch# monitor capture mycap export flash:mycap1.pcap
Exported Successfully
```

Step 14: Check that the file has been created and that it contains the packets by entering:

```
Switch# dir flash:mycap1.pcap
Directory of flash:/mycap1.pcap

14758  -rw-       20152  Apr 15 2012 16:00:28 -07:00  mycap1.pcap

831541248 bytes total (831340544 bytes free)
Switch# show monitor capture file flash:mycap1.pcap brief
  1   0.000000     10.1.1.2 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2   1.000000     10.1.1.3 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3   2.000000     10.1.1.4 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4   3.000000     10.1.1.5 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5   4.000000     10.1.1.6 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6   5.000000     10.1.1.7 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7   6.000000     10.1.1.8 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8   7.000000     10.1.1.9 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9   8.000000    10.1.1.10 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 10   9.000000    10.1.1.11 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 11  10.000000    10.1.1.12 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 12  11.000000    10.1.1.13 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 13  12.000000    10.1.1.14 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

```
 14  13.000000    10.1.1.15 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

 15  14.000000    10.1.1.16 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

 16  15.000000    10.1.1.17 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

Step 15: Stop the packet capture and display the buffer contents by entering:

```
Switch# monitor capture mycap stop
Switch# show monitor capture mycap buffer brief

  0.000000     10.1.1.2 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000     10.1.1.3 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000     10.1.1.4 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000     10.1.1.5 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000     10.1.1.6 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000     10.1.1.7 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000     10.1.1.8 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000     10.1.1.9 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.10 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.11 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 10.000000    10.1.1.12 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 11.000000    10.1.1.13 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

Step 16: Clear the buffer and then try to display packets from the buffer by entering:

```
Switch# monitor capture mycap clear
Switch# show monitor capture mycap buffer brief
```

Step 17: Delete the capture point by entering:

```
Switch# no monitor capture mycap
```

Example: Capture Sessions

```
Switch# monitor capture mycap start display display-filter "stp"
  0.000000 20:37:06:cf:08:b6 -> 01:80:c2:00:00:00 STP Conf. Root = 32768/100/20:37:06:ce:f0:80
   Cost = 0  Port = 0x8136
  2.000992 20:37:06:cf:08:b6 -> 01:80:c2:00:00:00 STP Conf. Root = 32768/100/20:37:06:ce:f0:80
   Cost = 0  Port = 0x8136
   2.981996 20:37:06:cf:08:b6 -> 20:37:06:cf:08:b6 LOOP Reply
  4.000992 20:37:06:cf:08:b6 -> 01:80:c2:00:00:00 STP Conf. Root = 32768/100/20:37:06:ce:f0:80
   Cost = 0  Port = 0x8136
  6.000000 20:37:06:cf:08:b6 -> 01:80:c2:00:00:00 STP Conf. Root = 32768/100/20:37:06:ce:f0:80
   Cost = 0  Port = 0x8136
  7.998001 20:37:06:cf:08:b6 -> 01:80:c2:00:00:00 STP Conf. Root = 32768/100/20:37:06:ce:f0:80
   Cost = 0  Port = 0x8136
  9.998001 20:37:06:cf:08:b6 -> 01:80:c2:00:00:00 STP Conf. Root = 32768/100/20:37:06:ce:f0:80
   Cost = 0  Port = 0x8136
Capture test is not active Failed to Initiate Wireshark
Switch# show monitor capture mycap parameter
   monitor capture mycap control-plane both
   monitor capture mycap match any
   monitor capture mycap file location flash:mycap1.1 buffer-size 90
   monitor capture mycap limit duration 10

Switch# monitor capture mycap start display display-filter "udp.port == 20002"
A file by the same capture file name already exists, overwrite?[confirm] [ENTER]
after a minute or so...
Capture mycap is not active Failed to Initiate Wireshark
*Oct 13 15:00:44.649: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
*Oct 13 15:00:46.657: %BUFCAP-6-DISABLE_ASYNC: Capture Point mycap disabled. Rea
son : Wireshark Session Ended

Switch# monitor capture mycap start display display-filter "udp.port == 20002" dump
A file by the same capture file name already exists, overwrite?[confirm]
after a minute or so...
Capture mycap is not active Failed to Initiate Wireshark
*Oct 13 15:00:44.649: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
```

```
*Oct 13 15:00:46.657: %BUFCAP-6-DISABLE_ASYNC: Capture Point mycap disabled. Rea
son : Wireshark Session Ended

Switch# no monitor capture mycap file
Switch# monitor capture mycap start display display-filter "udp.port == 20002" dump
 Please associate capture file/buffer
Unable to activate Capture.

Switch# monitor capture mycap start display display-filter "udp.port == 20002"
 Please associate capture file/buffer
Unable to activate Capture.

Switch# monitor capture mycap start display detailed
 Please associate capture file/buffer
Unable to activate Capture.
```

Example: Capture and Store in Lock-step Mode

This example captures live traffic and stores the packets in lock-step mode.

Note The capture rate might be slow for the first 15 seconds. If possible and necessary, start the traffic 15 seconds after the capture session starts.

Step 1: Define a capture point to match on the relevant traffic and associate it to a file by entering:

```
Switch# monitor capture mycap interface GigabitEthernet1/0/1 in
Switch# monitor capture mycap match ipv4 any any
Switch# monitor capture mycap limit duration 60 packets 100
Switch# monitor capture mycap file location flash:mycap.pcap buffer-size 64
```

Step 2: Confirm that the capture point has been correctly defined by entering:

```
Switch# show monitor capture mycap parameter
   monitor capture mycap interface GigabitEthernet1/0/1 in
   monitor capture mycap file location flash:mycap.pcap buffer-size 64
   monitor capture mycap limit packets 100 duration 60

Switch# show monitor capture mycap

Status Information for Capture mycap
  Target Type:
   Interface: GigabitEthernet1/0/1, Direction: in
   Status : Inactive
  Filter Details:
   Filter not attached
  Buffer Details:
   Buffer Type: LINEAR (default)
  File Details:
   Associated file name: flash:mycap.pcap
   Size of buffer(in MB): 64
  Limit Details:
   Number of Packets to capture: 100
   Packet Capture duration: 60
   Packet Size to capture: 0 (no limit)
   Packets per second: 0 (no limit)
   Packet sampling rate: 0 (no sampling)
```

Step 3: Launch packet capture by entering:

```
Switch# monitor capture mycap start
A file by the same capture file name already exists, overwrite?[confirm]
Turning on lock-step mode

Switch#
*Oct 14 09:35:32.661: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
```

Step 4: Display the packets by entering:

```
Switch# show monitor capture file flash:mycap.pcap
  0.000000    10.1.1.30 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000    10.1.1.31 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000    10.1.1.32 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000    10.1.1.33 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000    10.1.1.34 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000    10.1.1.35 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000    10.1.1.36 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000    10.1.1.37 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.38 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.39 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

Step 5: Delete the capture point by entering:

```
Switch# no monitor capture mycap
```

Example: Simple Capture and Store of Packets in Egress Direction

This example shows how to capture packets to a filter:

Step 1: Define a capture point to match on the relevant traffic and associate it to a file by entering:

```
Switch# monitor capture mycap interface Gigabit 1/0/1 out match ipv4 any any
Switch# monitor capture mycap limit duration 60 packets 100
Switch# monitor capture mycap file location flash:mycap.pcap buffer-size 90
```

Step 2: Confirm that the capture point has been correctly defined by entering:

```
Switch# show monitor capture mycap parameter
   monitor capture mycap interface GigabitEthernet1/0/1 out
   monitor capture mycap match ipv4  any any
   monitor capture mycap file location flash:mycap.pcap buffer-size 90
   monitor capture mycap limit packets 100 duration 60

Switch# show monitor capture mycap

Status Information for Capture mycap
  Target Type:
   Interface: GigabitEthernet1/0/1, Direction: out
   Status : Inactive
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  Buffer Details:
   Buffer Type: LINEAR (default)
  File Details:
   Associated file name: flash:mycap.pcap
   Size of buffer(in MB): 90
  Limit Details:
   Number of Packets to capture: 100
   Packet Capture duration: 60
   Packet Size to capture: 0 (no limit)
   Packets per second: 0 (no limit)
   Packet sampling rate: 0 (no sampling)
```

Step 3: Launch packet capture by entering:

```
Switch# monitor capture mycap start
A file by the same capture file name already exists, overwrite?[confirm]
Turning on lock-step mode

Switch#
*Oct 14 09:35:32.661: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
```

Note Allow the capture operation stop automatically after the time has elapsed or the packet count has been met. When you see the following message in the output, will know that the capture operation has stopped:

```
*Oct 14 09:36:34.632: %BUFCAP-6-DISABLE_ASYNC: Capture Point mycap disabled. Rea
son : Wireshark Session Ended
```

The mycap.pcap file now contains the captured packets.

Step 4: Display the packets by entering:

```
Switch# show monitor capture file flash:mycap.pcap
  0.000000    10.1.1.30 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000    10.1.1.31 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000    10.1.1.32 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000    10.1.1.33 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000    10.1.1.34 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000    10.1.1.35 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000    10.1.1.36 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000    10.1.1.37 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.38 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.39 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

Step 5: Delete the capture point by entering:

```
Switch# no monitor capture mycap
```

Additional References

Related Documents

Related Topic	Document Title
General Packet Filtering	For general packet filtering, refer to: Display Filter Reference

Error Message Decoder

Description	Link
To help you research and resolve system error messages in this release, use the Error Message Decoder tool.	https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

Standards and RFCs

Standard/RFC	Title

MIBs

MIB	MIBs Link
All supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Related Topics

Filters, on page 130

Feature History and Information for WireShark

Release	Modification
Cisco IOS XE 3.3SE	This feature was introduced.

INDEX

S

T

U

V