Cisco Nexus 3550-T Configuration Guide, Release 10.1(x)

First Published: 2021-12-17

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

CONTENTS

PART I

Cisco Nexus 3550-T Configuration Guide Overview

CHAPTER 1

Preface

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

This preface includes the following sections:

Full Cisco Trademarks with Software License

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

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

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

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

Audience

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

Document Conventions

Command descriptions use the following conventions:

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

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

Examples use the following conventions:

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

Related Documentation for Cisco Nexus® 3550-T Switches

The entire Cisco Nexus® 3550-T switch documentation set is available at the following URL:

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

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus3550t-docfeedback@cisco.com. We appreciate your feedback.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.

- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

CHAPTER 2

Configuration Overview

Overview of the Cisco Nexus® 3550-T Switches

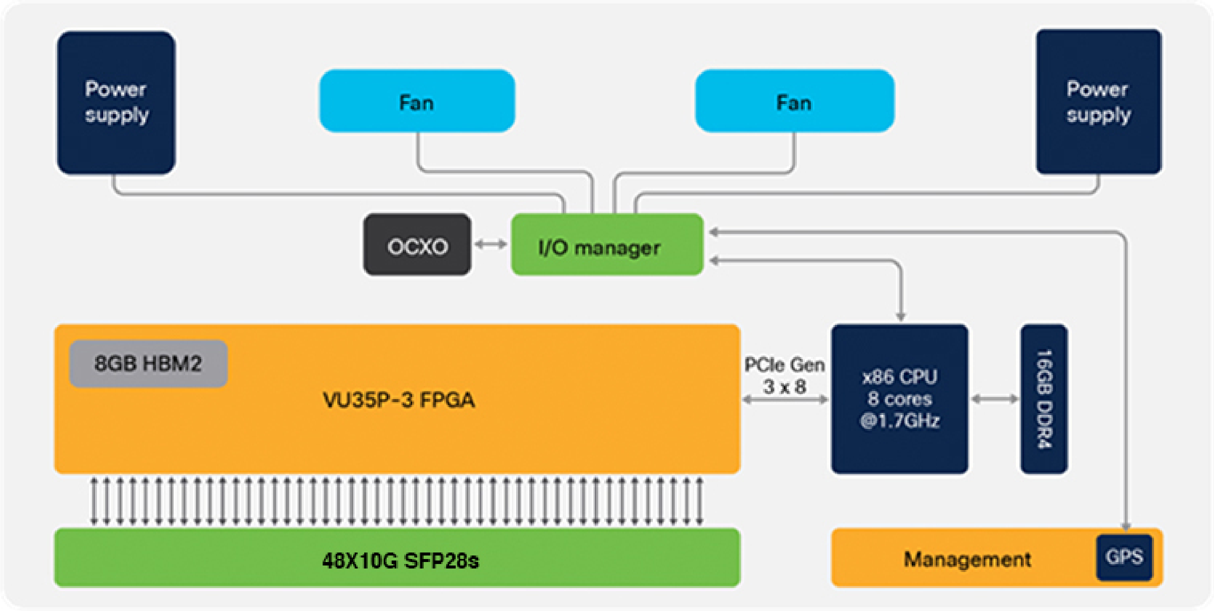
The Cisco Nexus® 3550-T Programmable Network Platform is a top-of-rack software application platform with a unique low-latency design.

A flexible low-latency application platform

The Cisco Nexus® 3550-T Platform features up to 48 ports of Ethernet connectivity in a single-rack unit form factor. The platform is built around a powerful programmable FPGA and provides a complete firmware development environment for custom application and use cases.

A block diagram of the Cisco Nexus® 3550-T Platform is given below.

Figure 1: Data sheet Cisco public



Cisco Nexus® 3550-T Programmable Network Platform

Benefits

- Cisco Nexus® 3550-T next-generation ultra-low latency network switch platforms are specifically designed to address the needs of latency sensitive applications in data center networking, High-Frequency Trading (HFT), financial services, and service provider networks.
- Cisco Nexus® 3550-T Platforms and Switches Ultra-low latency switch platform, FPGA application programming, multiplexing and precision timestamping to facilitate your mission-critical network applications.

Unlock the full value of your ULL network solutions

Cisco Nexus® 3550-T next-generation ultra-low latency network switch platforms are specifically designed to address the needs of latency sensitive applications in data center networking and service provider networks.

Cisco Nexus 3550-T Series Platforms and Switches Ultra-low latency switch platform, FPGA application programming, multiplexing and precision timestamping to facilitate your mission-critical network applications.

Cisco environmental sustainability

Information about Cisco's environmental sustainability policies and initiatives for our products, solutions, operations, and extended operations or supply chain is provided in the "Environment Sustainability" section of Cisco's Corporate Social Responsibility (CSR) Report.

Reference links to information about key environmental sustainability topics (mentioned in the "Environment Sustainability" section of the CSR Report) are provided in the following table:

Sustainability Topic	Reference
Information on product material content laws and regulations	Materials
Information on electronic waste laws and regulations, including products, batteries, and packaging	WEEE compliance

Cisco makes the packaging data available for informational purposes only. It may not reflect the most current legal developments, and Cisco does not represent, warrant, or guarantee that it is complete, accurate, or up to date. This information is subject to change without notice.

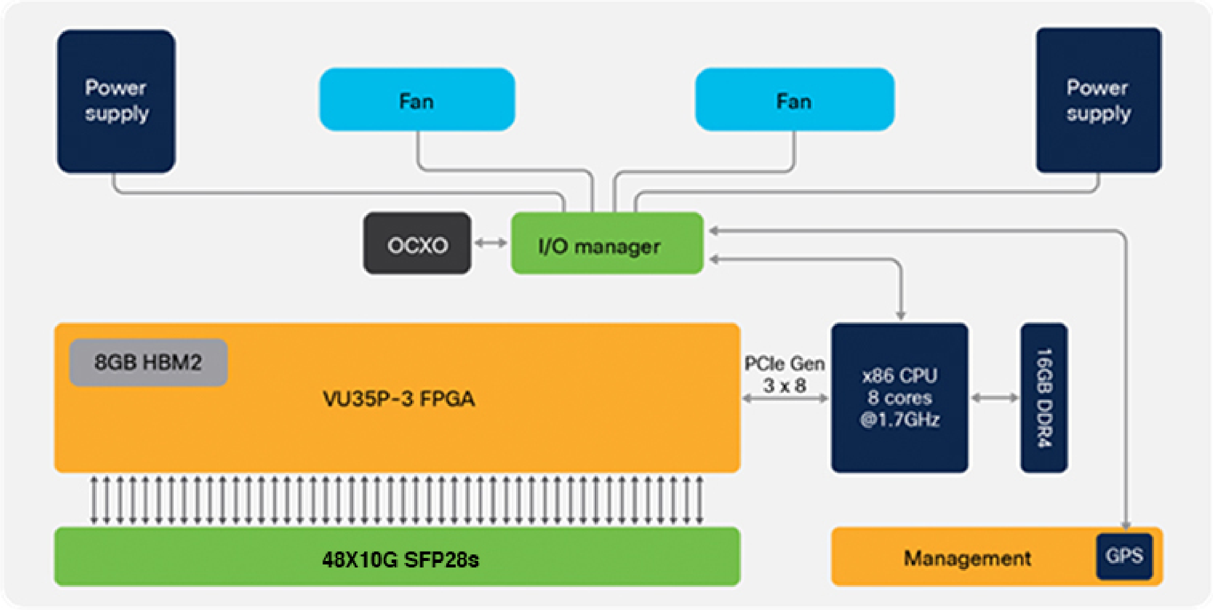
Flexible payment solutions to help you achieve your objectives

Cisco Capital® makes it easier to get the right technology to achieve your objectives, enable business transformation, and help you stay competitive. We can help you reduce the total cost of ownership, conserve capital, and accelerate growth. In more than 100 countries, our flexible payment solutions can help you acquire hardware, software, services and complementary third-party equipment in easy, predictable payments. Learn more.

Hardware Architecture of the Cisco Nexus® 3550-T Switches

The Cisco Nexus® 3550-T Programmable Network Platform has a fixed form factor that is built around a dynamically reconfigurable FPGA (Field Programmable Gate Array) and provides 48 ports that are 10G capable along with an x86 (Intel® Atom® processor with 8 cores up to 1.7 GHz)–management CPU. All 48 ports are directly connected to Xilinx Virtex UltraScale Plus VU35P FPGA with a “-3” speed grade. The FPGA has 8GB of High Bandwidth Memory (HBM) on board. The Cisco Nexus® 3550-T hardware architecture diagram is shown in Figure 2 below.

Figure 2: Cisco Nexus 3550-T Programmable Network Platform Data Sheet



Cisco Nexus® 3550-T Programmable Network Platform hardware architecture

Ease of Management

The Cisco Nexus® 3550-T Programmable Network Platform features a console port, a Micro USB port, a 1G RJ45 port, and a 10G SFP+ port, which can be used as management interfaces.

The Cisco Nexus® 3550-T Programmable Network Platform includes standard enterprise manageability and deployment capability features.

Programmability

The Cisco Nexus® 3550-T Programmable Network Platform provides a powerful development framework to add application-specific intelligence to the Cisco Nexus® 3550-T FPGA Module.

Switch platform features

The Cisco Nexus® 3550-T Programmable Network Platform supports packet-aware statistics. The Cisco Nexus® 3550-T firmware has the capability to monitor for vital packet statistics, including the number of packets/bytes transmitted/received and transmit/receive errors, and deep diagnostics, including light levels, operating temperatures, transceiver capabilities, and more.

All these statistics are available at no latency cost on the critical path. The following are some of the available statistics:

Connectivity

- 48 x SFP28 (small form-factor pluggable) configuration (backward compatible with SFP+ and SFP)
- SFP+ Fiber (10GBASE-SR, 10GBASE-LR, 10GBASE-LRM, 1000BASE-SX, 1000BASE-LX)
- SFP+ Copper Direct Attach

- RJ45 management port
- SMA for PPS in/out* (3.3V with 50 Ohm signal interface)
- SMA for GPS* in
- RJ45 management port
- RJ45 industry-standard serial port (default speed: 115200 N81)
- USB (for firmware upgrades)

General

- 19” 1RU, rack mount
- Weight 10kg (22lbs)
- Dual, hot-swappable supplies
- Standard: AC 90-264V, 47-64 Hz, included IEC C13-C14 cables
- Optional: DC 40-72V
- Maximum consumption: 150W
- Dual hot-swappable fan modules
- Optional airflow direction
- Operating temperature: -5 °C to 45 °C
- Storage temperature: -40 °C to 70 °C
- Operating Relative Humidity: 5% to 90% (noncondensing)
- Storage Relative Humidity: 5% to 95% (noncondensing)

PART II

Cisco Nexus 3550-T Security Configuration Guide

CHAPTER 3

Security Overview

The Cisco NX-OS software supports security features that can protect your network against degradation or failure and also against data loss or compromise resulting from intentional attacks and from unintended but damaging mistakes by well-meaning network users.

This chapter includes the following sections:

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Authentication, Authorization, and Accounting

Authentication, authorization, and accounting (AAA) is an architectural framework for configuring a set of three independent security functions in a consistent, modular manner.

Authentication

Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods and then applying that list to various interfaces.

Authorization

Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.

Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared with the information contained in a database for a given user, and the result is returned to AAA to determine the user's actual capabilities and restrictions.

Accounting
Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes. Accounting enables you to track the services that users are accessing, as well as the amount of network resources that they are consuming.

Note You can configure authentication outside of AAA. However, you must configure AAA if you want to use RADIUS or TACACS+, or if you want to configure a backup authentication method.

For more information, see the Configuring AAA, on page 17 chapter.

RADIUS and TACACS+ Security Protocols

AAA uses security protocols to administer its security functions. If your router or access server is acting as a network access server, AAA is the means through which you establish communication between your network access server and your RADIUS security server.

The chapters in this guide describe how to configure the following security server protocols:

RADIUS
A distributed client/server system implemented through AAA that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

TACACS+
A security application implemented through AAA that provides a centralized validation of users who are attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. TACACS+ provides for separate and modular authentication, authorization, and accounting facilities.

For more information, see the Configuring RADIUS, on page 51 chapter.

SSH and Telnet

You can use the Secure Shell (SSH) server to enable an SSH client to make a secure, encrypted connection to a Cisco NX-OS device. SSH uses strong encryption for authentication. The SSH server in the Cisco NX-OS software can interoperate with publicly and commercially available SSH clients.

The SSH client in the Cisco NX-OS software works with publicly and commercially available SSH servers.

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site and then passes the keystrokes from one device to the other. Telnet can accept either an IP address or a domain name as the remote device address.

For more information, see the Configuring SSH and Telnet, on page 97 chapter.

IP ACLs

IP ACLs are ordered sets of rules that you can use to filter traffic based on IPv4 information in the Layer 3 header of packets. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the Cisco NX-OS software determines that an IP ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether a packet is permitted or denied, or if there is no match, the Cisco NX-OS software applies the applicable default rule. The Cisco NX-OS software continues processing packets that are permitted and drops packets that are denied.

For more information, see the Configuring IP ACLs, on page 79 chapter.

CHAPTER 4

Configuring AAA

This chapter describes how to configure authentication, authorization, and accounting (AAA) on Cisco NX-OS devices.

This chapter includes the following sections:

About AAA

This section includes information about AAA on Cisco NX-OS devices.

AAA Security Services

The AAA feature allows you to verify the identity of, grant access to, and track the actions of users managing a Cisco NX-OS device. Cisco NX-OS devices support Remote Access Dial-In User Service (RADIUS) or Terminal Access Controller Access Control System Plus (TACACS+) protocols.

Based on the user ID and password combination that you provide, Cisco NX-OS devices perform local authentication or authorization using the local database or remote authentication or authorization using one or more AAA servers. A preshared secret key provides security for communication between the Cisco NX-OS device and AAA servers. You can configure a common secret key for all AAA servers or for only a specific AAA server.

AAA security provides the following services:

Authentication

Identifies users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption.

Authentication is the process of verifying the identity of the person or device accessing the Cisco NX-OS device, which is based on the user ID and password combination provided by the entity trying to access the Cisco NX-OS device. Cisco NX-OS devices allow you to perform local authentication (using the local lookup database) or remote authentication (using one or more RADIUS or TACACS+ servers).

Authorization

Provides access control.AAA authorization is the process of assembling a set of attributes that describe what the user is authorized to perform. Authorization in the Cisco NX-OS software is provided by attributes that are downloaded from AAA servers. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights with the appropriate user.

Accounting

Provides the method for collecting information, logging the information locally, and sending the information to the AAA server for billing, auditing, and reporting.

The accounting feature tracks and maintains a log of every management session used to access the Cisco NX-OS device. You can use this information to generate reports for troubleshooting and auditing purposes. You can store accounting logs locally or send them to remote AAA servers.

Note The Cisco NX-OS software supports authentication, authorization, and accounting independently. For example, you can configure authentication and authorization without configuring accounting.

Benefits of Using AAA

AAA provides the following benefits:

- Increased flexibility and control of access configuration
- Scalability
- Standardized authentication methods, such as RADIUS and TACACS+
- Multiple backup devices

Remote AAA Services

Remote AAA services provided through RADIUS and TACACS+ protocols have the following advantages over local AAA services:

- It is easier to manage user password lists for each Cisco NX-OS device in the fabric.
- AAA servers are already deployed widely across enterprises and can be easily used for AAA services.
- You can centrally manage the accounting log for all Cisco NX-OS devices in the fabric.
- It is easier to manage user attributes for each Cisco NX-OS device in the fabric than using the local databases on the devices.

AAA Server Groups

You can specify remote AAA servers for authentication, authorization, and accounting using server groups. A server group is a set of remote AAA servers that implements the same AAA protocol. The purpose of a server group is to provide for failover servers in case a remote AAA server fails to respond. If the first remote server in the group fails to respond, the next remote server in the group is tried until one of the servers sends a response. If all the AAA servers in the server group fail to respond, then that server group option is considered a failure. If required, you can specify multiple server groups. If the Cisco NX-OS device encounters errors from the servers in the first group, it tries the servers in the next server group.

AAA Service Configuration Options

The AAA configuration in Cisco NX-OS devices is service based, which means that you can have separate AAA configurations for the following services:

- User Telnet or Secure Shell (SSH) login authentication
- Console login authentication
- Extensible Authentication Protocol over User Datagram Protocol (EAPoUDP) authentication for Network Admission Control (NAC)
- User management session accounting

This table provides the related CLI command for each AAA service configuration option.

Table 1: AAA Service Configuration Commands

AAA Service Configuration Option	Related Command
Telnet or SSH login	**aaa authentication login default**
Console login	**aaa authentication login console**
	aaa authentication eou default
User session accounting	**aaa accounting default**

You can specify the following authentication methods for the AAA services:

All RADIUS servers

Uses the global pool of RADIUS servers for authentication.

Specified server groups

Uses specified RADIUS, TACACS+, or LDAP server groups you have configured for authentication.

Local

Uses the local username or password database for authentication.

None

Specifies that no AAA authentication be used.

Note If you specify the all RADIUS servers method, rather than a specified server group method, the Cisco NX-OS device chooses the RADIUS server from the global pool of configured RADIUS servers, in the order of configuration. Servers from this global pool are the servers that can be selectively configured in a RADIUS server group on the Cisco NX-OS device.

This table shows the AAA authentication methods that you can configure for the AAA services.

Table 2: AAA Authentication Methods for AAA Services

AAA Service	AAA Methods
Console login authentication	Server groups, local, and none
User login authentication	Server groups, local, and none
User management session accounting	Server groups and local

Note For console login authentication, user login authentication, and user management session accounting, the Cisco NX-OS device tries each option in the order specified. The local option is the default method when other configured options fail. You can disable the local option for the console or default login by using the **no aaa authentication login** {**console** | **default**} **fallback error local** command.

Authentication and Authorization Process for User Login

The following list explains the process:

- When you log in to the required Cisco NX-OS device, you can use the Telnet, SSH, or console login options.
- When you have configured the AAA server groups using the server group authentication method, the Cisco NX-OS device sends an authentication request to the first AAA server in the group as follows:
  - If the AAA server fails to respond, the next AAA server is tried and so on until the remote server responds to the authentication request.
  - If all AAA servers in the server group fail to respond, the servers in the next server group are tried.
  - If all configured methods fail, the local database is used for authentication, unless fallback to local is disabled for the console login.
- If the Cisco NX-OS device successfully authenticates you through a remote AAA server, then the following possibilities apply:
  - If the AAA server protocol is RADIUS, then user roles specified in the cisco-av-pair attribute are downloaded with an authentication response.
  - If the AAA server protocol is TACACS+, then another request is sent to the same server to get the user roles specified as custom attributes for the shell.

- If your username and password are successfully authenticated locally, the Cisco NX-OS device logs you in and assigns you the roles configured in the local database.

Note "No more server groups left" means that there is no response from any server in all server groups. "No more servers left" means that there is no response from any server within this server group.

AES Password Encryption and Primary Encryption Keys

You can enable strong, reversible 128-bit Advanced Encryption Standard (AES) password encryption, also known as type-6 encryption. To start using type-6 encryption, you must enable the AES password encryption feature and configure a primary encryption key, which is used to encrypt and decrypt passwords.

After you enable AES password encryption and configure a primary key, all existing and newly created clear-text passwords for supported applications (currently RADIUS and TACACS+) are stored in type-6 encrypted format, unless you disable type-6 password encryption. You can also configure Cisco NX-OS to convert all existing weakly encrypted passwords to type-6 encrypted passwords.

Prerequisites for AAA

Remote AAA servers have the following prerequisites:

- Ensure that at least one RADIUS, TACACS+, or LDAP server is reachable through IP.
- Ensure that the Cisco NX-OS device is configured as a client of the AAA servers.
- Ensure that the secret key is configured on the Cisco NX-OS device and the remote AAA servers.
- Ensure that the remote server responds to AAA requests from the Cisco NX-OS device.

Guidelines and Limitations for AAA

AAA has the following guidelines and limitations:

- If you have a user account that is configured on the local Cisco NX-OS device that has the same name as a remote user account on an AAA server, the Cisco NX-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.
- Cisco Nexus® 3550-T switches support the **aaa authentication login ascii-authentication** command only for TACACS+ (and not for RADIUS).
- If you modify the default login authentication method (without using the **local** keyword), the configuration overrides the console login authentication method. To explicitly configure the console authentication method, use the **aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**} command.
- The **login block-for** and **login quiet-mode** configuration mode commands are renamed to **system login block-for** and **system login quiet-mode**, respectively.

- When you use the **system login quiet-mode access-class QUIET_LIST** command, you must ensure that the access list is correctly defined to only block the specified traffic. For example, if you need to block only the user logins from untrusted hosts, then the access list should specify ports 22, 23, 80, and 443 corresponding to SSH, telnet, and HTTP-based access from those hosts.

Default Settings for AAA

This table lists the default settings for AAA parameters.

Table 3: Default AAA Parameter Settings

Parameters	Default
Console authentication method	local
Default authentication method	local
Login authentication failure messages	Disabled
CHAP authentication	Disabled
MSCHAP authentication	Disabled
Default accounting method	local
Accounting log display length	250 KB

Configuring AAA

This section describes the tasks for configuring AAA on Cisco NX-OS devices.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Note Cisco Nexus® 3550-T Series switches support the CLI command, aaa authentication login ascii-authentication, only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors.

Process for Configuring AAA

Follow these steps to configure AAA authentication and accounting:

1. If you want to use remote RADIUS, TACACS+, or LDAP servers for authentication, configure the hosts on your Cisco NX-OS device.
2. Configure console login authentication methods.
3. Configure default login authentication methods for user logins.
4. Configure default AAA accounting default methods.

Configuring Console Login Authentication Methods

This section describes how to configure the authentication methods for the console login.

The authentication methods include the following:

- Global pool of RADIUS servers
- Named subset of RADIUS, TACACS+, or LDAP servers
- Local database on the Cisco NX-OS device
- Username only (none)

The default method is local, but you have the option to disable it.

Note The **group radius** and **group** *server-name* forms of the **aaa authentication** command refer to a set of previously defined RADIUS servers. Use the **radius-server host** command to configure the host servers. Use the **aaa group server radius** command to create a named group of servers.

Note If you perform a password recovery when remote authentication is enabled, local authentication becomes enabled for console login as soon as the password recovery is done. As a result, you can log into the Cisco NX-OS device through the console port using the new password. After login, you can continue to use local authentication, or you can enable remote authentication after resetting the admin password configured at the AAA servers. For more information about the password recovery process, see the *Cisco Nexus® Series NX-OS Troubleshooting Guide.*

Before you begin

Configure RADIUS, TACACS+, or LDAP server groups, as needed.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**} **Example:** `switch(config)# aaa authentication login console group radius`	Configures login authentication methods for the console. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: **radius** Uses the global pool of RADIUS servers for authentication. ***named-group*** Uses a named subset of RADIUS, TACACS+, or LDAP servers for authentication. The **local** method uses the local database for authentication, and the **none** method specifies that no AAA authentication be used. The default console login method is **local**, which is used when no methods are configured or when all the configured methods fail to respond, unless fallback to local is disabled for the console login.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show aaa authentication** **Example:** `switch# show aaa authentication`	Displays the configuration of the console login authentication methods.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Default Login Authentication Methods

The authentication methods include the following:

- Global pool of RADIUS servers
- Named subset of RADIUS, TACACS+, or LDAP servers
- Local database on the Cisco NX-OS device
- Username only

The default method is local, but you have the option to disable it.

Before you begin

Configure RADIUS, TACACS+, or LDAP server groups, as needed.

Procedure

	Command or Action	Purpose

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

Step 2	**aaa authentication login default** {**group** *group-list* [**none**] | **local** | **none**} **Example:** `switch(config)# aaa authentication login default group radius`	Configures the default authentication methods. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius**—Uses the global pool of RADIUS servers for authentication. • *named-group*—Uses a named subset of RADIUS, TACACS+, or LDAP servers for authentication. The **local** method uses the local database for authentication, and the **none** method specifies that no AAA authentication be used. The default login method is **local**, which is used when no methods are configured or when all the configured methods fail to respond, unless fallback to local is disabled for the console login. You can configure one of the following: • AAA authentication groups • AAA authentication groups with no authentication • Local authentication • No authentication

	Command or Action	Purpose
		Note The **local** keyword is not supported (and is not required) when configuring AAA authentication groups because local authentication is the default if remote servers are unreachable. For example, if you configure **aaa authentication login default group g1**, local authentication is tried if you are unable to authenticate using AAA group g1. In contrast, if you configure **aaa authentication login default group g1 none**, no authentication is performed if you are unable to authenticate using AAA group g1.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show aaa authentication** **Example:** `switch# show aaa authentication`	Displays the configuration of the default login authentication methods.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Disabling Fallback to Local Authentication

By default, if remote authentication is configured for console or default login and all AAA servers are unreachable (resulting in an authentication error), the Cisco NX-OS device falls back to local authentication to ensure that users aren’t locked out of the device. However, you can disable fallback to local authentication in order to increase security.

Caution Disabling fallback to local authentication can lock your Cisco NX-OS device, forcing you to perform a password recovery in order to gain access. To prevent being locked out of the device, we recommend that you disable fallback to local authentication for only the default login or the console login, not both.

Before you begin

Configure remote authentication for the console or default login.

Procedure

	Command or Action	Purpose

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

Step 2	**no aaa authentication login** {**console** | **default**} **fallback error local** **Example:** `switch(config)# no aaa authentication login console fallback error local`	Disables fallback to local authentication for the console or default login if remote authentication is configured and all AAA servers are unreachable. The following message appears when you disable fallback to local authentication: `“WARNING!!! Disabling fallback can lock your switch.”`

Step 3	(Optional) **exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show aaa authentication** **Example:** `switch# show aaa authentication`	Displays the configuration of the console and default login authentication methods.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling the Default User Role for AAA Authentication

You can allow remote users who do not have a user role to log in to the Cisco NX-OS device through a RADIUS or TACACS+ remote authentication server using a default user role. When you disable the AAA default user role feature, remote users who do not have a user role cannot log in to the device.

Procedure

	Command or Action	Purpose

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

Step 2	**aaa user default-role** **Example:** `switch(config)# aaa user default-role`	Enables the default user role for AAA authentication. The default is enabled. You can disable the default user role feature by using the **no** form of this command.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show aaa user default-role** **Example:** `switch# show aaa user default-role`	Displays the AAA default user role configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling Login Authentication Failure Messages

When you log in, the login is processed by rolling over to the local user database if the remote AAA servers do not respond. In such cases, the following messages display on the user's terminal if you have enabled login failure messages:

```
Remote AAA servers unreachable; local authentication done.
```

```
Remote AAA servers unreachable; local authentication failed.
```

Procedure

	Command or Action	Purpose

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

Step 2	**aaa authentication login error-enable** **Example:**	Enables login authentication failure messages. The default is disabled.

	Command or Action	Purpose
	`switch(config)# aaa authentication login error-enable`	

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show aaa authentication** **Example:** `switch# show aaa authentication`	Displays the login failure message configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Logging Successful and Failed Login Attempts

You can configure the switch to log all successful and failed login attempts to the configured syslog server.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	Required: [**no**] **login on-failure log** **Example:** `switch(config)# login on-failure log`	Logs all failed authentication messages to the configured syslog server only if the logging level is set to 6. With this configuration, the following syslog message appears after the failed login: AUTHPRIV-3-SYSTEM_MSG: pam_aaa:Authentication failed for user admin from 172.22.00.00 **Note** When logging level authpriv is 6, additional Linux kernel authentication messages appear along with the previous message. If these additional messages need to be ignored, the authpriv value should be set to 3.

	Command or Action	Purpose

Step 3	Required: [**no**] **login on-success log**
	Example:
	`switch(config)# login on-success log`
	`switch(config)# logging level authpriv 6`
	`switch(config)# logging level daemon 6`
		Logs all successful authentication messages to the configured syslog server only if the logging level is set to 6. With this configuration, the following syslog message appears after the successful login:
		AUTHPRIV-6-SYSTEM_MSG: pam_aaa:Authentication success for user admin from 172.22.00.00
		Note When logging level authpriv is 6, additional Linux kernel authentication messages appear along with the previous message. If these additional messages need to be ignored, the authpriv value should be set to 3.

Step 4	(Optional) **show login on-failure log** **Example:** `switch(config)# show login on-failure log`	Displays whether the switch is configured to log failed authentication messages to the syslog server.
Step 5	(Optional) **show login on-successful log** **Example:** `switch(config)# show login on-successful log`	Displays whether the switch is configured to log successful authentication messages to the syslog server.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Login Block Per User

Ensure that the switch is in global configuration mode.

The Login Block Per User feature helps detect suspected Denial of Service (DoS) attacks and to slow down dictionary attacks. This feature is applicable for local users and remote users. Use this task to configure login parameters to block a user after failed login attempts.

Note You can configure login block for remote users.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**aaa authentication rejected** *attempts***in***seconds***ban***seconds* **Example:** `switch(config)# aaa authentication rejected 3 in 20 ban 300`	Configures login parameters to block a user. **Note** Use **no aaa authentication rejected** command to revert to the default login parameters.
Step 3	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 4	(Optional) **show running config** **Example:** `switch# show running config`	Displays the login parameters.
Step 5	**show aaa local user blocked** **Example:** `switch# show aaa local user blocked`	Displays the blocked local users.
Step 6	**clear aaa local user blocked {username user| all}** **Example:** `switch(config)# switch# clear aaa local user blocked username testuser`	Clears the blocked local users. all –Clears all the blocked local users.
Step 7	**show aaa user blocked** **Example:** `switch(config)# show aaa user blocked`	Displays all blocked local and remote users.
Step 8	(Optional) **clear aaa user blocked{username user| all}** **Example:** `switch# clear aaa user blocked username testuser`	Clears all blocked local and remote users. all – Clears all the blocked local and remote users.

Example

Note Only network-admin have privileges to run the show and clear commands.

The following example shows how to configure the login parameters to block a user for 300 seconds when three login attempts fail within a period of 20 seconds:

```
switch(config)# aaa authentication rejected 3 in 20 ban 300
switch# show run | i rejected
aaa authentication rejected 3 in 20 ban 300
switch# show aaa local user blocked
Local-user                State
testuser                  Watched (till 11:34:42 IST Nov 12 2020)
switch# clear aaa local user blocked username testuser
switch# show aaa user blocked
Local-user                State
testuser                  Watched (till 11:34:42 IST Nov 12 2020)
switch# clear aaa user blocked username testuser
```

Enabling CHAP Authentication

The Cisco NX-OS software supports the Challenge Handshake Authentication Protocol (CHAP), a challenge-response authentication protocol that uses the industry-standard Message Digest (MD5) hashing scheme to encrypt responses. You can use CHAP for user logins to a Cisco NX-OS device through a remote authentication server (RADIUS or TACACS+).

By default, the Cisco NX-OS device uses Password Authentication Protocol (PAP) authentication between the Cisco NX-OS device and the remote server. If you enable CHAP, you need to configure your RADIUS or TACACS+ server to recognize the CHAP vendor-specific attributes (VSAs).

Note Cisco Nexus® 3550-T switches support the CLI command, aaa authentication login ascii-authentication, only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors.

This table shows the RADIUS and TACACS+ VSAs required for CHAP.

Table 4: CHAP RADIUS and TACACS+ VSAs

Vendor-ID Number	Vendor-Type Number	VSA	Description
311	11	CHAP-Challenge	Contains the challenge sent by an AAA server to a CHAP user. It can be used in both Access-Request and Access-Challenge packets.
211	11	CHAP-Response	Contains the response value provided by a CHAP user in response to the challenge. It is used only in Access-Request packets.

Before you begin

Disable AAA ASCII authentication for logins.

Procedure

	Command or Action	Purpose

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

Step 2	**no aaa authentication login ascii-authentication** **Example:** `switch(config)# `**`no aaa authentication login ascii-authentication`**	Disables ASCII authentication.
Step 3	**aaa authentication login chap enable** **Example:** `switch(config)# `**`aaa authentication login chap enable`**	Enables CHAP authentication. The default is disabled. **Note** You cannot enable both CHAP and MSCHAP or MSCHAP V2 on your Cisco NX-OS device.

Step 4	(Optional) **exit**
	Example:
	`switch(config)# `**`exit`**
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show aaa authentication login chap** **Example:** `switch# `**`show aaa authentication login chap`**	Displays the CHAP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# `**`copy running-config startup-config`**	Copies the running configuration to the startup configuration.

Enabling MSCHAP or MSCHAP V2 Authentication

Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is the Microsoft version of CHAP. The Cisco NX-OS software also supports MSCHAP Version 2 (MSCHAP V2). You can use MSCHAP for user logins to a Cisco NX-OS device through a remote authentication server (RADIUS or TACACS+). MSCHAP V2 only supports user logins to a Cisco NX-OS device through remote authentication RADIUS servers. If you configure a TACACS+ group with MSCHAP V2, the AAA default login authentication uses the next configured method, or the local method, if no other server group is configured.

Note The Cisco NX-OS software may display the following message:

“ Warning: MSCHAP V2 is supported only with Radius.”

This warning message is informational only and does not affect MSCHAP V2 operation with RADIUS.

By default, the Cisco NX-OS device uses Password Authentication Protocol (PAP) authentication between the Cisco NX-OS device and the remote server. If you enable MSCHAP or MSCHAP V2, you need to configure your RADIUS server to recognize the MSCHAP and MSCHAP V2 vendor-specific attributes (VSAs).

This table shows the RADIUS VSAs required for MSCHAP.

Table 5: MSCHAP and MSCHAP V2 RADIUS VSAs

Vendor-ID Number	Vendor-Type Number	VSA	Description
311	11	MSCHAP-Challenge	Contains the challenge sent by an AAA server to an MSCHAP or MSCHAP V2 user. It can be used in both Access-Request and Access-Challenge packets.
211	11	MSCHAP-Response	Contains the response value provided by an MSCHAP or MSCHAP V2 user in response to the challenge. It is only used in Access-Request packets.

Before you begin

Disable AAA ASCII authentication for logins.

Procedure

	Command or Action	Purpose

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

Step 2	**no aaa authentication login ascii-authentication** **Example:** `switch(config)# no aaa authentication login ascii-authentication`	Disables ASCII authentication.
Step 3	**aaa authentication login** {**mschap** | **mschapv2**} **enable** **Example:** `switch(config)# aaa authentication login mschap enable`	Enables MSCHAP or MSCHAP V2 authentication. The default is disabled. **Note** You cannot enable both MSCHAP and MSCHAP V2 on your Cisco NX-OS device.

	Command or Action	Purpose

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show aaa authentication login** {**mschap** | **mschapv2**} **Example:** `switch# show aaa authentication login mschap`	Displays the MSCHAP or MSCHAP V2 configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring AAA Accounting Default Methods

Cisco NX-OS software supports TACACS+ and RADIUS methods for accounting. Cisco NX-OS devices report user activity to TACACS+ or RADIUS security servers in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the AAA server.

When you activate AAA accounting, the Cisco NX-OS device reports these attributes as accounting records, which are then stored in an accounting log on the security server.

You can create default method lists defining specific accounting methods, which include the following:

RADIUS server group
Uses the global pool of RADIUS servers for accounting.

Specified server group
Uses a specified RADIUS or TACACS+ server group for accounting.

Local
Uses the local username or password database for accounting.

Note If you have configured server groups and the server groups do not respond, by default, the local database is used for authentication.

Before you begin

Configure RADIUS or TACACS+ server groups, as needed.

Procedure

	Command or Action	Purpose

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

Step 2	**aaa accounting default** {**group** *group-list* | **local**}
	Example:
	`switch(config)# aaa accounting default group radius`
		Configures the default accounting method.
		The *group-list* argument consists of a space-delimited list of group names. The group names are the following:
		• **radius**—Uses the global pool of RADIUS servers for accounting.
		• *named-group*—Uses a named subset of TACACS+ or RADIUS servers for accounting.
		The **local** method uses the local database for accounting.
		The default method is **local**, which is used when no server groups are configured or when all the configured server groups fail to respond.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show aaa accounting** **Example:** `switch# show aaa accounting`	Displays the configuration AAA accounting default methods.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Using AAA Server VSAs with Cisco NX-OS Devices

You can use vendor-specific attributes (VSAs) to specify Cisco NX-OS user roles and SNMPv3 parameters on AAA servers.

About VSAs

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating VSAs between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Cisco attribute for a particular type of authorization, the separator is = (equal sign) for mandatory attributes, and * (asterisk) indicates optional attributes.

When you use RADIUS servers for authentication on a Cisco NX-OS device, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

VSA Format

The following VSA protocol options are supported by the Cisco NX-OS software:

Shell

Protocol used in access-accept packets to provide user profile information.

Accounting

Protocol used in accounting-request packets. If a value contains any white spaces, put it within double quotation marks.

The following attributes are supported by the Cisco NX-OS software:

roles

Lists all the roles assigned to the user. The value field is a string that stores the list of group names delimited by white space. For example, if you belong to role network-operator and network-admin, the value field would be network-operator network-admin. This subattribute is sent in the VSA portion of the Access-Accept frames from the RADIUS server, and it can only be used with the shell protocol value. These examples use the roles attribute:

```
shell:roles=network-operator network-admin
```

```
shell:roles*network-operator network-admin
```

The following examples show the roles attribute as supported by FreeRADIUS:

```
Cisco-AVPair = shell:roles=\network-operator network-admin\
```

```
Cisco-AVPair = shell:roles*\network-operator network-admin\
```

Note When you specify a VSA as shell:roles*"network-operator network-admin" or "shell:roles*\"network-operator network-admin\"", this VSA is flagged as an optional attribute and other Cisco devices ignore this attribute.

accountinginfo

Stores accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch, and it can only be used with the accounting protocol-related PDUs.

Specifying Cisco NX-OS User Roles and SNMPv3 Parameters on AAA Servers

You can use the VSA cisco-av-pair on AAA servers to specify user role mapping for the Cisco NX-OS device using this format:

```
shell:roles="roleA roleB …"
```

If you do not specify the role option in the cisco-av-pair attribute, the default user role is network-operator.

You can also specify your SNMPv3 authentication and privacy protocol attributes as follows:

```
shell:roles="roleA roleB..." snmpv3:auth=SHA priv=AES-128
```

The SNMPv3 authentication protocol options are SHA and MD5. The privacy protocol options are AES-128 and DES. If you do not specify these options in the cisco-av-pair attribute, MD5 and DES are the default authentication protocols.

Configuring Secure Login Features

Configuring Login Parameters

You can configure login parameters to automatically block further login attempts when a possible denial-of-service (DoS) attack is detected and slow down dictionary attacks by enforcing a quiet period if multiple failed connection attempts are detected.

Note This feature restarts if a system switchover occurs or the AAA process restarts.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	[**no**] **login block-for** *seconds* **attempts** *tries* **within** *seconds* **Example:** `switch(config)# login block-for 100 attempts 2 within 60`	Configures the quiet mode time period. The range for all arguments is from 1 to 65535. The example shows how to configure the switch to enter a 100-second quiet period if 2 failed login attempts are exceeded within 60 seconds. After you enter this command, all login attempts made through Telnet or SSH are denied during the quiet period. Access control lists (ACLs)

	Command or Action	Purpose
		are not exempt from the quiet period until the command is entered. **Note** You must enter this command before any other login command can be used.
Step 3	(Optional) [**no**] **login quiet-mode access-class** *acl-name* **Example:** `switch(config)#  login quiet-mode access-class myacl`	Specifies an ACL that is to be applied to the switch when it changes to quiet mode. When the switch is in quiet mode, all login requests are denied, and the only available connection is through the console.
Step 4	(Optional) **show login** [**failures**] **Example:** `switch(config)# show  login`	Displays the login parameters. The **failures** option displays information related only to failed login attempts.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Restricting User Login Sessions

You can restrict the maximum number of simultaneous login sessions per user. Doing so prevents users from having multiple unwanted sessions and solves the potential security issue of unauthorized users accessing a valid SSH or Telnet session.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	[**no**] **user max-logins** *max-logins* **Example:** `switch(config)# user max-logins 1`	Restricts the maximum number of simultaneous login sessions per user. The range is from 1 to 7. If you set the maximum login limit as 1, only one Telnet or SSH session is allowed per user. **Note** The configured login limit applies to all users. You cannot set a different limit for individual users.
Step 3	(Optional) **show running-config all | i max-login** **Example:**	Displays the maximum number of login sessions allowed per user.

	Command or Action	Purpose
	`switch(config)# show running-config all | i max-login`	
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Restricting the Password Length

You can restrict the minimum and maximum length of the user password. This feature enables you to increase system security by forcing the user to provide a strong password.

Before you begin

You must enable password strength checking using the **password strength-check** command. If you restrict the password length but do not enable password strength checking and the user enters a password that is not within the restricted length, an error appears, but a user account is created. To enforce the password length and prevent a user account from being created, you must enable password strength checking and restrict the password length.

Procedure

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

Step 2	[**no**] **userpassphrase** {**min-length** *min-length* | **max-length** *max-length*}
	Example:
	`switch(config)# userpassphrase min-length 8 max-length 80`
		Restricts the minimum and/or maximum length of the user password. The minimum password length is from 4 to 127 characters, and the maximum password length is from 80 to 127 characters.

Step 3	(Optional) **show userpassphrase** {**length** | **max-length** | **min-length**}
	Example:
	`switch(config)# show userpassphrase length`
		Displays the minimum and maximum length of the user password.

Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling the Password Prompt for the Username

You can configure the switch to prompt the user to enter a password after entering the username.

Procedure

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

Step 2	**password prompt username**
	Example:
	`switch(config)# password prompt username`
	`Password prompt username is enabled.`
	`After providing the required options in`
	` the username command, press enter.`
	`User will be prompted for the username`
	`password and password will be hidden.`
	`Note: Choosing password key in the same`
	` line while configuring user account,`
	`password will not be hidden.`
		Configures the switch to prompt the user to enter a password after she enters the **username** command without the **password** option or the **snmp-server user** command. The password that the user enters will be hidden. You can use the **no** form of this command to disable this feature.

Step 3	(Optional) **copy running-config startup-config**
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Configuring the Shared Secret for RADIUS or TACACS+

The shared secret that you configure for remote authentication and accounting between the switch and the RADIUS or TACACS+ server should be hidden because it is sensitive information. You can use a separate command to generate an encrypted shared secret for the **radius-server** [**host**] **key** and **tacacs-server** [**host**] **key** commands. The SHA256 hashing method is used to store the encrypted shared secret.

Procedure

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

Step 2	**generate type7_encrypted_secret**
	Example:
	`switch(config)# generate`
	`type7_encrypted_secret`
	`Type-7 (Vigenere) Encryption,`
	` Use this encrypted secret to configure`
		Configures the RADIUS or TACACS+ shared secret with key type 7. You are prompted to enter the shared secret in plain text twice. The secret is hidden as you enter it. Then an encrypted version of the secret appears.

	Command or Action	Purpose

	` radius and tacacs shared secret with`
	`key type 7.`
	` Copy complete secret with double quotes.`
	
	`Enter plain text secret:`
	`Confirm plain text secret:`
	`Type 7 Encrypted secret is : "fewhg"`
		Note You can generate the encrypted equivalent of a plain-text secret separately and configure the encrypted shared secret later using the **radius-server** [**host**] **key** and **tacacs-server** [**host**] **key** commands.

Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Monitoring and Clearing the Local AAA Accounting Log

The Cisco NX-OS device maintains a local log for the AAA accounting activity. You can monitor this log and clear it.

Procedure

	Command or Action	Purpose
Step 1	**show accounting log** [*size* | **last-index** | **start-seqnum** *number* | **start-time** *year month day hh*:*mm*:*ss*] **Example:** `switch# show accounting log`	Displays the accounting log contents. By default, the command output contains up to 250,000 bytes of the accounting log. You can use the *size* argument to limit command output. The range is from 0 to 250000 bytes. You can also specify a starting sequence number or a starting time for the log output.The range of the starting index is from 1 to 1000000. Use the **last-index** keyword to display the value of the last index number in the accounting log file.
Step 2	(Optional) **clear accounting log** [**logflash**] **Example:** `switch# clear aaa accounting log`	Clears the accounting log contents. The **logflash** keyword clears the accounting log stored in the logflash.

Verifying the AAA Configuration

To display AAA configuration information, perform one of the following tasks:

Command	Purpose
show aaa accounting	Displays AAA accounting configuration.

Command	Purpose
show aaa authentication [**login** {**ascii-authentication** | **chap** | **error-enable** | **mschap** | **mschapv2**}]	Displays AAA authentication login configuration information.
show aaa groups	Displays the AAA server group configuration.
show login [**failures**]	Displays the login parameters. The **failures** option displays information related only to failed login attempts. **Note** The **clear login failures** command clears the login failures in the current watch period.
show login on-failure log	Displays whether the switch is configured to log failed authentication messages to the syslog server.
show login on-successful log	Displays whether the switch is configured to log successful authentication messages to the syslog server.
show running-config aaa [**all**]	Displays the AAA configuration in the running configuration.
show running-config all | i max-login	Displays the maximum number of login sessions allowed per user.
show startup-config aaa	Displays the AAA configuration in the startup configuration.
show userpassphrase {**length** | **max-length** | **min-length**}	Displays the minimum and maximum length of the user password.

Configuration Examples for AAA

The following example shows how to configure AAA:

```
aaa authentication login default group radius
aaa authentication login console group radius
aaa accounting default group radius
```

Configuration Examples for Login Parameters

The following example shows how to configure the switch to enter a 100-second quiet period if 3 failed login attempts is exceeded within 60 seconds. This example shows no login failures.

```
switch# configure terminal
switch(config)# login block-for 100 attempts 3 within 60
switch(config)# show login

No Quiet-Mode access list has been configured, default ACL will be applied.

 Switch is enabled to watch for login Attacks.
 If more than 3 login failures occur in 60 seconds or less,
 logins will be disabled for 100 seconds.

 Switch presently in Normal-Mode.
 Current Watch Window remaining time 45 seconds.
 Present login failure count 0.

switch(config)# show login failures
*** No logged failed login attempts with the device.***
```

The following example shows how to configure a quiet-mode ACL. All login requests are denied during the quiet period except hosts from the myacl ACL. This example also shows a login failure.

```
switch# configure terminal
switch(config)# login block-for 100 attempts 3 within 60
switch(config)# login quiet-mode access-class myacl

switch(config)# show login

 Switch is enabled to watch for login Attacks.
 If more than 3 login failures occur in 60 seconds or less,
 logins will be disabled for 100 seconds.

 Switch presently in Quiet-Mode.
 Will remain in Quiet-Mode for 98 seconds.
 Denying logins from all sources.

switch(config)# show login failures
Information about last 20 login failure's with the device.
--------------------------------------------------------------------------------
Username      Line          SourceIPAddr    Appname    TimeStamp
--------------------------------------------------------------------------------
asd           /dev/pts/0    171.70.55.158   login      Mon Aug  3 18:18:54 2015
qweq          /dev/pts/0    171.70.55.158   login      Mon Aug  3 18:19:02 2015
qwe           /dev/pts/0    171.70.55.158   login      Mon Aug  3 18:19:08 2015
--------------------------------------------------------------------------------
```

Configuration Examples for the Password Prompt Feature

The following example shows how to configure the switch to prompt the user to enter a password after she enters the **username** command and the error message that displays if she does not enter a password.

```
switch# configure terminal
switch(config)# password prompt username
Password prompt username is enabled.
```

```
After providing the required options in the username command, press enter.
User will be prompted for the username password and password will be hidden.
Note: Choosing password key in the same line while configuring user account, password will
 not be hidden.


switch(config)# username user1
Enter password:
Confirm password:
warning: password for user:user1 not set. S/he may not be able to login
```

The following example shows how to configure the switch to prompt the user to enter a password after she enters the **snmp-server user** command and the prompts that then display to the user.

```
switch# configure terminal
switch(config)# password prompt username
Password prompt username is enabled.
After providing the required options in the username command, press enter.
User will be prompted for the username password and password will be hidden.
Note: Choosing password key in the same line while configuring user account, password will
 not be hidden.


N3550-T(config)# snmp-server user user1
Enter auth md5 password (Press Enter to Skip):
Enter auth sha password (Press Enter to Skip):
```

Additional References for AAA

This section includes additional information related to implementing AAA.

Related Documents

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
MIBs related to AAA	To locate and download supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

5 장

Configuring Traffic Storm Control

This chapter describes how to configure traffic storm control on the Cisco NX-OS device.

There is no Control Plane Policing (CoPP) implemented in the Cisco Nexus® 3550-T switches hardware. Storm-Control can be used to control the amount of traffic to CPU from each port. Storm-control feature on the Cisco Nexus® 3550-T switches does not provide any traffic classification.

Licensing Requirements for Traffic Storm Control

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Traffic storm control requires no license. Any feature not included in a license package is bundled with the nx-os image and is provided at no extra charge to you. For an explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Guidelines and Limitations for Traffic Storm Control

Traffic storm control has the following configuration guidelines and limitations:

- Cisco Nexus® 3550-T switches support setting the maximum allowed traffic frame rate per interface.

  This is provided through the CLI via the following command:

  ```
  storm-control-cpu all <rate>
  ```

- The pps range can be from 0 to 250000000.
- The default value is 2000.
- Storm control is applicable for physical interfaces, both L2 and L3.
- It is supported on port-channels.

The following limitations apply to Cisco Nexus® 3550-T switches:

- This rate limiting applies for all traffic types i.e., no protocol/pkt-type based throttling.
    - Due to the above limitation, it is possible that control packets such as CDP, LACP, ARP, OSPF can be lost when there are a number of software forwarded data packets, cache-misses or mac-learn notifications.

Configuration Examples for Traffic Storm Control

The following example shows how to configure traffic storm control:

```
module-1# show hardware internal exbl hw port_dump 46
port 46:
  unknown mcast      : false   l3 mcast           : false
  dst nat            : false   src nat            : false
  in meta            : false   out meta           : false
  mux mode           : false   access mode        : true
  default route      : false   forwarding         : true
  bridge (l2)        : true    ucast fail to sw   : false
  dst acl            : false   src acl            : false
  vrrp               : false

  rate burst size    : 255
  rate delay (cycles): 125000
<snip>
```

```
switch(config)# sh run int e1/47
!Command: show running-config interface Ethernet1/47
!Running configuration last done at: Thu Jan 31 01:10:45 2008
!Time: Thu Jan 31 19:01:15 2008

version 10.1(2t)E1(0) Bios:version 3.2

interface Ethernet1/47
  switchport
  no shutdown
```

```
module-1# show hardware internal exbl hw port_dump 46
port 46:
  unknown mcast      : false   l3 mcast           : false
  dst nat            : false   src nat            : false
  in meta            : false   out meta           : false
  mux mode           : false   access mode        : true
  default route      : false   forwarding         : true
  bridge (l2)        : true    ucast fail to sw   : false
  dst acl            : false   src acl            : false
  vrrp               : false

  rate burst size    : 255
  rate delay (cycles): 50000
<snip>
```

```
switch(config)# sh run int e1/47
!Command: show running-config interface Ethernet1/47
!Running configuration last done at: Thu Jan 31 01:10:45 2008
!Time: Thu Jan 31 19:01:15 2008

version 10.1(2t)E1(0) Bios:version 3.2

interface Ethernet1/47
```

```
  switchport
  storm-control-cpu all 5000
  no shutdown

module-1# show hardware internal exbl hw port_dump 46
port 46:
  unknown mcast      : false   l3 mcast           : false
  dst nat            : false   src nat            : false
  in meta            : false   out meta           : false
  mux mode           : false   access mode        : true
  default route      : false   forwarding         : true
  bridge (l2)        : true    ucast fail to sw   : false
  dst acl            : false   src acl            : false
  vrrp               : false

  rate burst size    : 255
  rate delay (cycles): 1
<snip>

switch(config)# sh run int e1/47
!Command: show running-config interface Ethernet1/47
!Running configuration last done at: Thu Jan 31 01:10:45 2008
!Time: Thu Jan 31 19:01:15 2008

version 10.1(2t)E1(0) Bios:version 3.2

interface Ethernet1/47
  switchport
  storm-control-cpu all 250000000
  no shutdown

module-1# show hardware internal exbl hw port_dump 46
port 46:
  unknown mcast      : false   l3 mcast           : false
  dst nat            : false   src nat            : false
  in meta            : false   out meta           : false
  mux mode           : false   access mode        : true
  default route      : false   forwarding         : true
  bridge (l2)        : true    ucast fail to sw   : false
  dst acl            : false   src acl            : false
  vrrp               : false

  rate burst size    : 255
  rate delay (cycles): 0  --□ Drops all packtes to host/CPU from this port
<snip>

switch(config)# sh run int e1/47
!Command: show running-config interface Ethernet1/47
!Running configuration last done at: Thu Jan 31 01:10:45 2008
!Time: Thu Jan 31 19:01:15 2008

version 10.1(2t)E1(0) Bios:version 3.2

interface Ethernet1/47
  switchport
  storm-control-cpu all 0
  no shutdown

module-1# show hardware internal exbl hw port_dump 46
port 46:
  unknown mcast      : false   l3 mcast           : false
  dst nat            : false   src nat            : false
  in meta            : false   out meta           : false
  mux mode           : false   access mode        : true
  default route      : false   forwarding         : true
  bridge (l2)        : true    ucast fail to sw   : false
  dst acl            : false   src acl            : false
  vrrp               : false

  rate burst size    : 255
  rate delay (cycles): 0  --□ Drops all packtes to host/CPU from this port
<snip>
```

```
switch(config)# sh run int e1/47
!Command: show running-config interface Ethernet1/47
!Running configuration last done at: Thu Jan 31 01:10:45 2008
!Time: Thu Jan 31 19:01:15 2008

version 10.1(2t)E1(0) Bios:version 3.2

interface Ethernet1/47
  switchport
  storm-control-cpu all 0
  no shutdown
```

Additional References for Traffic Storm Control

This section includes additional information related to implementing traffic storm control.

Related Documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*

CHAPTER 6

Configuring RADIUS

This chapter describes how to configure the Remote Access Dial-In User Service (RADIUS) protocol on Cisco NX-OS devices.

This chapter includes the following sections:

About RADIUS

The RADIUS distributed client/server system allows you to secure networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco NX-OS devices and send authentication and accounting requests to a central RADIUS server that contains all user authentication and network service access information.

RADIUS Network Environments

RADIUS can be implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

You can use RADIUS in the following network environments that require access security:

- Networks with multiple-vendor network devices, each supporting RADIUS. For example, network devices from several vendors can use a single RADIUS server-based security database.

- Networks already using RADIUS. You can add a Cisco NX-OS device with RADIUS to the network. This action might be the first step when you make a transition to a AAA server.
- Networks that require resource accounting. You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider (ISP) might use a freeware-based version of the RADIUS access control and accounting software to meet special security and billing needs.
- Networks that support authentication profiles. Using the RADIUS server in your network, you can configure AAA authentication and set up per-user profiles. Per-user profiles enable the Cisco NX-OS device to better manage ports using their existing RADIUS solutions and to efficiently manage shared resources to offer different service-level agreements.

RADIUS Operation

When a user attempts to log in and authenticate to a Cisco NX-OS device using RADIUS, the following process occurs:

- The user is prompted for and enters a username and password.
- The username and encrypted password are sent over the network to the RADIUS server.
- The user receives one of the following responses from the RADIUS server:

  ACCEPT
  The user is authenticated.
  REJECT
  The user is not authenticated and is prompted to reenter the username and password, or access is denied.
  CHALLENGE
  A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
  CHANGE PASSWORD
  A request is issued by the RADIUS server, asking the user to select a new password.

The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 address, access list, and user timeouts.

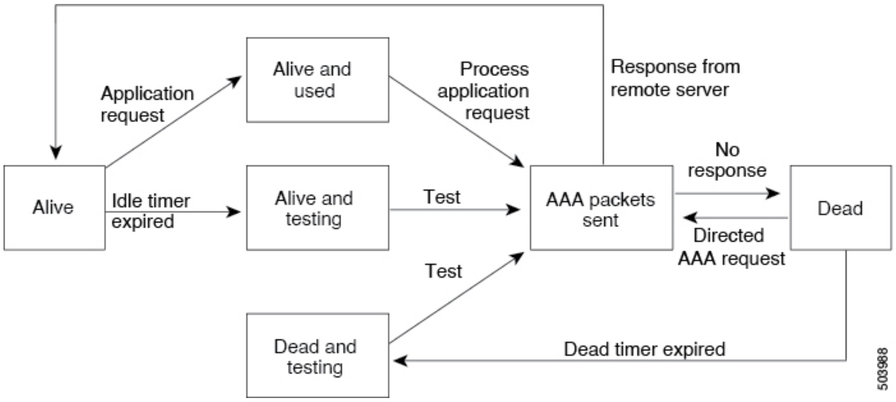
RADIUS Server Monitoring

An unresponsive RADIUS server can cause a delay in processing AAA requests. You can configure the Cisco NX-OS device to periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco NX-OS device marks unresponsive RADIUS servers as dead and does not send AAA requests to any dead RADIUS servers. The Cisco NX-OS device periodically monitors the dead RADIUS servers and brings them to the alive state once they respond. This monitoring process verifies that a RADIUS server is in a working state before real AAA requests are sent its way. Whenever a

RADIUS server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the Cisco NX-OS device displays an error message that a failure is taking place.

Figure 3: RADIUS Server States

This figure shows the states for RADIUS server monitoring.



Note The monitoring interval for alive servers and dead servers are different and can be configured by the user. The RADIUS server monitoring is performed by sending a test authentication request to the RADIUS server.

Vendor-Specific Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating VSAs between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named cisco-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Cisco attribute for a particular type of authorization, the separator is = (equal sign) for mandatory attributes, and * (asterisk) indicates optional attributes.

When you use RADIUS servers for authentication on a Cisco NX-OS device, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, with authentication results. This authorization information is specified through VSAs.

The following VSA protocol options are supported by the Cisco NX-OS software:

Shell
Protocol used in access-accept packets to provide user profile information.

Accounting
Protocol used in accounting-request packets. If a value contains any white spaces, you should enclose the value within double quotation marks.

The Cisco NX-OS software supports the following attributes:

roles

Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space. For example, if the user belongs to roles network-operator and network-admin, the value field would be network-operator network-admin. This subattribute, which the RADIUS server sends in the VSA portion of the Access-Accept frames, can only be used with the shell protocol value. The following examples show the roles attribute that is supported by the Cisco Access Control Server (ACS):

```
shell:roles=network-operator network-admin
```

```
shell:roles*“network-operator network-admin
```

The following examples show the roles attribute that is supported by FreeRADIUS:

```
Cisco-AVPair = shell:roles=\network-operator network-admin\
```

```
Cisco-AVPair = shell:roles*\network-operator network-admin\
```

Note When you specify a VSA as shell:roles*"network-operator network-admin" or "shell:roles*\"network-operator network-admin\"", this VSA is flagged as an optional attribute and other Cisco devices ignore this attribute.

accountinginfo

Stores accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch. It can be used only with the accounting protocol data units (PDUs).

About RADIUS Change of Authorization

A standard RADIUS interface is typically used in a pulled model, in which the request originates from a device attached to a network and the response is sent from the queried servers. Cisco NX-OS sofware supports the RADIUS Change of Authorization (CoA) request defined in RFC 5176 that is used in a pushed model, in which the request originates from the external server to the device attached to the network, and enables the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

When Dot1x is enabled, the network device acts as the authenticator and is responsible for processing dynamic COA per session.

The following requests are supported:

- Session reauthentication
- Session termination

Session Reauthentication

To initiate session reauthentication, the authentication, authorization, and accounting (AAA) server sends a standard CoA-Request message that contains a Cisco VSA and one or more session identification attributes. The Cisco VSA is in the form of Cisco:Avpair="subscriber:command=reauthenticate".

The current session state determines the response of the device to the message in the following scenarios:

- If the session is currently authenticated by IEEE 802.1x, the device responds by sending an Extensible Authentication Protocol over LAN (EAPOL)-RequestId message to the server.
- If the session is currently authenticated by MAC authentication bypass (MAB), the device sends an access request to the server, passing the same identity attributes used for the initial successful authentication.
- If session authentication is in progress when the device receives the command, the device terminates the process and restarts the authentication sequence, starting with the method configured to be attempted first.

Session Termination

A CoA Disconnect-Request terminates the session without disabling the host port. CoA Disconnect-Request termination causes reinitialization of the authenticator state machine for the specified host, but does not restrict the host's access to the network.

If the session cannot be located, the device returns a Disconnect-NAK message with the "Session Context Not Found" error-code attribute.

If the session is located, but the NAS was unable to remove the session due to some internal error, the device returns a Disconnect-NAK message with the "Session Context Not Removable" error-code attribute.

If the session is located, the device terminates the session. After the session has been completely removed, the device returns a Disconnect-ACK message.

Prerequisites for RADIUS

RADIUS has the following prerequisites:

- Obtain IPv4 addresses or hostnames for the RADIUS servers.
- Obtain keys from the RADIUS servers.
- Ensure that the Cisco NX-OS device is configured as a RADIUS client of the AAA servers.

Guidelines and Limitations for RADIUS

RADIUS has the following guidelines and limitations:

- You can configure a maximum of 64 RADIUS servers on the Cisco NX-OS device.

- If you have a user account configured on the local Cisco NX-OS device that has the same name as a remote user account on an AAA server, the Cisco NX-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.
- Only the RADIUS protocol supports one-time passwords.
- Cisco Nexus® 3550-T switches support the CLI command, aaa authentication login ascii-authentication, only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors.

Default Settings for RADIUS

This table lists the default settings for RADIUS parameters.

Table 6: Default RADIUS Parameter Settings

Parameters	Default
Server roles	Authentication and accounting
Dead timer interval	0 minutes
Retransmission count	1
Retransmission timer interval	5 seconds
Authentication port	1812
Accounting port	1813
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

Configuring RADIUS Servers

This section describes how to configure RADIUS servers on a Cisco NX-OS device.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Note Cisco Nexus® 3550-T switches support the CLI command, aaa authentication login ascii-authentication, only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors.

RADIUS Server Configuration Process

1. Establish the RADIUS server connections to the Cisco NX-OS device.
2. Configure the RADIUS secret keys for the RADIUS servers.
3. If needed, configure RADIUS server groups with subsets of the RADIUS servers for AAA authentication methods.
4. If needed, configure any of the following optional parameters:
   - Dead-time interval
   - RADIUS server specification allowed at user login
   - Timeout interval
   - TCP port
5. (Optional) If RADIUS distribution is enabled, commit the RADIUS configuration to the fabric.

Related Topics

Configuring RADIUS Server Hosts, on page 57
Configuring Global RADIUS Keys, on page 58

Configuring RADIUS Server Hosts

To access a remote RADIUS server, you must configure the IP address or hostname of a RADIUS server. You can configure up to 64 RADIUS servers.

Note By default, when you configure a RADIUS server IP address or hostname of the Cisco NX-OS device, the RADIUS server is added to the default RADIUS server group. You can also add the RADIUS server to another RADIUS server group.

Before you begin

Ensure that the server is already configured as a member of the server group.

Ensure that the server is configured to authenticate RADIUS traffic.

Ensure that the Cisco NX-OS device is configured as a RADIUS client of the AAA servers.

Procedure

	Command or Action	Purpose

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

Step 2	**radius-server host** {*ipv4-address* | *hostname*}
	Example:
	`switch(config)# radius-server host 10.10.1.1`
		Specifies the IPv4 address or hostname for a RADIUS server to use for authentication.

Step 3	(Optional) **show radius** {**pending** | **pending-diff**}
	Example:
	`switch(config)# show radius pending`
		Displays the RADIUS configuration pending for distribution.

Step 4	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 6	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring a Key for a Specific RADIUS Server, on page 59

Configuring Global RADIUS Keys

You can configure RADIUS keys for all servers used by the Cisco NX-OS device. A RADIUS key is a shared secret text string between the Cisco NX-OS device and the RADIUS server hosts.

Before you begin

Obtain the RADIUS key values for the remote RADIUS servers.

Configure the RADIUS key on the remote RADIUS servers.

Procedure

	Command or Action	Purpose

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

Step 2	**radius-server key** [**0** | **6** | **7**] *key-value* **Example:** `switch(config)# radius-server key 0 QsEfThUkO`	Specifies a RADIUS key for all RADIUS servers. You can specify that the *key-value* is in clear text format (**0**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters. By default, no RADIUS key is configured.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration. **Note** The RADIUS keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted RADIUS keys.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Groups, on page 61

Configuring a Key for a Specific RADIUS Server

You can configure a key on the Cisco NX-OS device for a specific RADIUS server. A RADIUS key is a secret text string shared between the Cisco NX-OS device and a specific RADIUS server.

Before you begin

Configure one or more RADIUS server hosts.

Obtain the key value for the remote RADIUS server.

Configure the key on the RADIUS server.

Procedure

	Command or Action	Purpose

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

Step 2	**radius-server host** {*ipv4-address* | *hostname*} **key** [**0** | **6** | **7**] *key-value* **Example:** `switch(config)# radius-server host 10.10.1.1 key 0 PlIjUhYg`	Specifies a RADIUS key for a specific RADIUS server. You can specify that the *key-value* is in clear text format (**0**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters. This RADIUS key is used instead of the global RADIUS key.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration. **Note** The RADIUS keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted RADIUS keys.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Hosts, on page 57

Configuring RADIUS Server Groups

You can specify one or more remote AAA servers for authentication using server groups. All members of a group must belong to the RADIUS protocol. The servers are tried in the same order in which you configure them.

You can configure these server groups at any time but they only take effect when you apply them to an AAA service.

Before you begin

Ensure that all servers in the group are RADIUS servers.

Procedure

	Command or Action	Purpose

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

Step 2	**aaa group server radius** *group-name*
	Example:
	`switch(config)# aaa group server radius RadServer`
	`switch(config-radius)#`
		Creates a RADIUS server group and enters the RADIUS server group configuration submode for that group. The *group-name* argument is a case-sensitive alphanumeric string with a maximum length of 127 characters.
		To delete a RADIUS server group, use the **no** form of this command.
		Note You are not allowed to delete the default system generated default group (RADIUS).

Step 3	**server** {*ipv4-address* | *hostname*}
	Example:
	`switch(config-radius)# server 10.10.1.1`
		Configures the RADIUS server as a member of the RADIUS server group.
		If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.

Step 4	(Optional) **deadtime** *minutes* **Example:** `switch(config-radius)# deadtime 30`	Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440. **Note** If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	(Optional) **server** {*ipv4-address* | *hostname*} **Example:**	Configures the RADIUS server as a member of the RADIUS server group.

	Command or Action	Purpose
	`switch(config-radius)# server 10.10.1.1`	**Tip** If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.
Step 6	(Optional) **use-vrf** *vrf-name* **Example:** `switch(config-radius)# use-vrf default`	Specifies the VRF to use to contact the servers in the server group.

Step 7	**exit**
	Example:
	`switch(config-radius)# exit`
	`switch(config)#`
		Exits configuration mode.

Step 8	(Optional) **show radius-server groups** [*group-name*] **Example:** `switch(config)# show radius-server groups`	Displays the RADIUS server group configuration.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring the RADIUS Dead-Time Interval, on page 71

Configuring the Global Source Interface for RADIUS Server Groups

You can configure a global source interface for RADIUS server groups to use when accessing RADIUS servers. You can also configure a different source interface for a specific RADIUS server group. By default, the Cisco NX-OS software uses any available interface.

Procedure

	Command or Action	Purpose

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

Step 2	**ip radius source-interface** *interface* **Example:** `switch(config)# ip radius source-interface mgmt 0`	Configures the global source interface for all RADIUS server groups configured on the device.

	Command or Action	Purpose

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration information.
Step 5	(Optional) **copy running-config startup config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Groups, on page 61

Allowing Users to Specify a RADIUS Server at Login

By default, the Cisco NX-OS device forwards an authentication request based on the default AAA authentication method. You can configure the Cisco NX-OS device to allow the user to specify a RADIUS server to send the authentication request by enabling the directed-request option. If you enable this option, the user can log in as *username*@**vrfname***hostname*, where **hostname** is the VRF to use and **hostname** is the name of a configured RADIUS server.

Note If you enable the directed-request option, the Cisco NX-OS device uses only the RADIUS method for authentication and not the default local method.

Note User-specified logins are supported only for Telnet sessions.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**radius-server directed-request** **Example:** `switch(config)# radius-server directed-request`	Allows users to specify a RADIUS server to send the authentication request when logging in. The default is disabled.

Step 3	(Optional) **show radius** {**pending** | **pending-diff**}
	Example:
	`switch(config)# show radius pending`
		Displays the RADIUS configuration pending for distribution.

Step 4	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 6	(Optional) **show radius-server directed-request** **Example:** `switch# show radius-server directed-request`	Displays the directed request configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Global RADIUS Transmission Retry Count and Timeout Interval

You can configure a global retransmission retry count and timeout interval for all RADIUS servers. By default, a Cisco NX-OS device retries transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. The timeout interval determines how long the Cisco NX-OS device waits for responses from RADIUS servers before declaring a timeout failure.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**radius-server retransmit** *count* **Example:** `switch(config)# radius-server retransmit 3`	Specifies the retransmission count for all RADIUS servers. The default retransmission count is 1 and the range is from 0 to 5.
Step 3	**radius-server timeout** *seconds* **Example:** `switch(config)# radius-server timeout 10`	Specifies the transmission timeout interval for RADIUS servers. The default timeout interval is 5 seconds and the range is from 1 to 60 seconds.

Step 4	(Optional) **show radius** {**pending** | **pending-diff**}
	Example:
	`switch(config)# show radius pending`
		Displays the RADIUS configuration pending for distribution.

Step 5	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration.

Step 6	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 7	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the RADIUS Transmission Retry Count and Timeout Interval for a Server

By default, a Cisco NX-OS device retries a transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. You can also set a timeout interval that the Cisco NX-OS device waits for responses from RADIUS servers before declaring a timeout failure.

Before you begin

Configure one or more RADIUS server hosts.

Procedure

	Command or Action	Purpose

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

Step 2	**radius-server host** {*ipv4-address* | *hostname*} **retransmit** *count* **Example:** `switch(config)# radius-server host server1 retransmit 3`	Specifies the retransmission count for a specific server. The default is the global value. **Note** The retransmission count value specified for a RADIUS server overrides the count specified for all RADIUS servers.
Step 3	**radius-server host** {*ipv4-address* | *hostname*} **timeout** *seconds* **Example:** `switch(config)# radius-server host server1 timeout 10`	Specifies the transmission timeout interval for a specific server. The default is the global value. **Note** The timeout interval value specified for a RADIUS server overrides the interval value specified for all RADIUS servers.

Step 4	(Optional) **show radius** {**pending** | **pending-diff**}
	Example:
	`switch(config)# show radius pending`
		Displays the RADIUS configuration pending for distribution.

Step 5	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration and distributes RADIUS configuration to other Cisco NX-OS devices if you have enabled CFS configuration distribution for the user role feature.

Step 6	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 7	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Hosts, on page 57

Configuring Accounting and Authentication Attributes for RADIUS Servers

You can specify that a RADIUS server is to be used only for accounting purposes or only for authentication purposes. By default, RADIUS servers are used for both accounting and authentication. You can also specify the destination UDP port numbers where RADIUS accounting and authentication messages should be sent if there is a conflict with the default port.

Before you begin

Configure one or more RADIUS server hosts.

Procedure

	Command or Action	Purpose

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

Step 2	(Optional) **radius-server host** {*ipv4-address* | *hostname*} **acct-port** *udp-port* **Example:** `switch(config)# radius-server host 10.10.1.1 acct-port 2004`	Specifies a UDP port to use for RADIUS accounting messages. The default UDP port is 1813. The range is from 0 to 65535.
Step 3	(Optional) **radius-server host** {*ipv4-address* | *hostname*} **accounting** **Example:** `switch(config)# radius-server host 10.10.1.1 accounting`	Specifies to use the RADIUS server only for accounting purposes. The default is both accounting and authentication.
Step 4	(Optional) **radius-server host** {*ipv4-address* | *hostname*} **auth-port** *udp-port* **Example:** `switch(config)# radius-server host 10.10.2.2 auth-port 2005`	Specifies a UDP port to use for RADIUS authentication messages. The default UDP port is 1812. The range is from 0 to 65535.
Step 5	(Optional) **radius-server host** {*ipv4-address* | *hostname*} **authentication** **Example:** `switch(config)# radius-server host 10.10.2.2 authentication`	Specifies to use the RADIUS server only for authentication purposes. The default is both accounting and authentication.

	Command or Action	Purpose

Step 6	(Optional) **show radius** {**pending** | **pending-diff**}
	Example:
	`switch(config)# show radius pending`
		Displays the RADIUS configuration pending for distribution.

Step 7	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration.

Step 8	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 9	(Optional) **show radius-server** **Example:** `switch(config)# show radius-server`	Displays the RADIUS server configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Hosts, on page 57

Configuring Global Periodic RADIUS Server Monitoring

You can monitor the availability of all RADIUS servers without having to configure the test parameters for each server individually. Any servers for which test parameters are not configured are monitored using the global level parameters.

Note Test parameters that are configured for individual servers take precedence over global test parameters.

The global configuration parameters include the username and password to use for the servers and an idle timer. The idle timer specifies the interval in which a RADIUS server receives no requests before the Cisco NX-OS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Note To protect network security, we recommend that you use a username that is not the same as an existing username in the RADIUS database.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic RADIUS server monitoring is not performed.

Before you begin

Enable RADIUS.

Procedure

	Command or Action	Purpose

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

Step 2	**radius-server test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}
	Example:
	`switch(config)# radius-server test username user1 password Ur2Gd2BH idle-time 3`
		Specifies parameters for global server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes.
		Note For periodic RADIUS server monitoring, the idle timer value must be greater than 0.

Step 3	**radius-server deadtime** *minutes* **Example:** `switch(config)# radius-server deadtime 5`	Specifies the number of minutes before the Cisco NX-OS device checks a RADIUS server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 0 to 1440 minutes.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring Periodic RADIUS Server Monitoring on Individual Servers, on page 70

Configuring Periodic RADIUS Server Monitoring on Individual Servers

You can monitor the availability of individual RADIUS servers. The configuration parameters include the username and password to use for the server and an idle timer. The idle timer specifies the interval during which a RADIUS server receives no requests before the Cisco NX-OS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Note Test parameters that are configured for individual servers take precedence over global test parameters.

Note For security reasons, we recommend that you do not configure a test username that is the same as an existing user in the RADIUS database.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, the Cisco NX-OS device does not perform periodic RADIUS server monitoring.

Before you begin

Enable RADIUS.

Add one or more RADIUS server hosts.

Procedure

	Command or Action	Purpose

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

Step 2	**radius-server host** {*ipv4-address* | *hostname*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}
	Example:
	`switch(config)# radius-server host 10.10.1.1 test username user1 password Ur2Gd2BH idle-time 3`
		Specifies parameters for individual server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes.
		Note For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.

Step 3	**radius-server deadtime** *minutes* **Example:** `switch(config)# radius-server deadtime 5`	Specifies the number of minutes before the Cisco NX-OS device checks a RADIUS server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 1 to 1440 minutes.

	Command or Action	Purpose

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.

Step 6	(Optional) **copy running-config startup-config**
	Example:
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Hosts, on page 57

Configuring Global Periodic RADIUS Server Monitoring, on page 68

Configuring the RADIUS Dead-Time Interval

You can configure the dead-time interval for all RADIUS servers. The dead-time interval specifies the time that the Cisco NX-OS device waits after declaring a RADIUS server is dead, before sending out a test packet to determine if the server is now alive. The default value is 0 minutes.

Note When the dead-time interval is 0 minutes, RADIUS servers are not marked as dead even if they are not responding. You can configure the dead-time interval for a RADIUS server group.

Procedure

	Command or Action	Purpose

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

Step 2	**radius-server deadtime** *minutes*
	Example:
	`switch(config)# radius-server deadtime`
	`5`
		Configures the dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes.

Step 3	(Optional) **show radius** {**pending** | **pending-diff**} **Example:**	Displays the RADIUS configuration pending for distribution.

	Command or Action	Purpose
	`switch(config)# show radius pending`	
Step 4	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 6	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Groups, on page 61

Configuring One-Time Passwords

One-time password (OTP) support is available for Cisco NX-OS devices through the use of RSA SecurID token servers. With this feature, users authenticate to a Cisco NX-OS device by entering both a personal identification number (or one-time password) and the token code being displayed at that moment on their RSA SecurID token.

Note The token code used for logging into the Cisco NX-OS device changes every 60 seconds. To prevent problems with device discovery, we recommend using different usernames that are present on the Cisco Secure ACS internal database.

Before you begin

On the Cisco NX-OS device, configure a RADIUS server host and remote default login authentication.

Ensure that the following are installed:

- Cisco Secure Access Control Server (ACS) version 4.2
- RSA Authentication Manager version 7.1 (the RSA SecurID token server)
- RSA ACE Agent/Client

No configuration (other than a RADIUS server host and remote authentication) is required on the Cisco NX-OS device to support one-time passwords. However, you must configure the Cisco Secure ACS as follows:

1. Enable RSA SecurID token server authentication.
2. Add the RSA SecurID token server to the Unknown User Policy database.

Manually Monitoring RADIUS Servers or Groups

You can manually issue a test message to a RADIUS server or to a server group.

Procedure

	Command or Action	Purpose
Step 1	**test aaa server radius** {*ipv4-address* | *hostname*} *username password* **Example:** `switch# test aaa server radius 10.10.1.1 user1 Ur2Gd2BH`	Sends a test message to a RADIUS server to confirm availability.
Step 2	**test aaa group** *group-name username password* **Example:** `switch# test aaa group RadGroup user2 As3He3CI`	Sends a test message to a RADIUS server group to confirm availability.

Enabling or Disabling Dynamic Author Server

Procedure

	Command or Action	Purpose

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

Step 2	**aaa server radius dynamic-author** **Example:** `switch(config)# aaa server radius dynamic-author`	Enables the RADIUS dynamic author server. You can disable the RADIUS dynamic author server using the no form of this command.

Configuring RADIUS Change of Authorization

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **aaa server radius dynamic-author** **Example:** `switch(config)# aaa server radius dynamic-author`	Configures the switch as an AAA server to facilitate interaction with an external policy server. You can disable the RADIUS dynamic author and the associated clients using the no form of this command.
Step 3	[**no**] **client** {*ip-address* | **hostname** } [**server-key** [**0** | **7**] *string*] **Example:** `switch(config-locsvr-da-radius)# client 192.168.0.5 server-key cisco1`	Configures the IP address or the hostname of the AAA server client. Use the optional server-key keyword and string argument to configure the server key at the client level. You can remove the client server using the no form of this command. **Note** Configuring the server key at the client level overrides the server key that is configured at the global level.
Step 4	[**no**] **port** *port-number* **Example:** `switch(config-locsvr-da-radius)# port 3799`	Specifies the port on which a device listens to the RADIUS requests from the configured RADIUS clients. The port range is 1 - 65535. You can revert to the default port using the no form of this command. **Note** The default port for a packet of disconnect is 1700.
Step 5	[**no**] **server-key** [**0** | **7**] *string*	Configures the global RADIUS key to be shared between a device and the RADIUS clients. You can remove the server-key using the no form of this command.

Verifying the RADIUS Configuration

To display RADIUS configuration information, perform one of the following tasks:

Command	Purpose
show radius {**status** | **pending** | **pending-diff**}	Displays the RADIUS Cisco Fabric Services distribution status and other details.

Command	Purpose
show running-config radius [**all**]	Displays the RADIUS configuration in the running configuration.
show startup-config radius	Displays the RADIUS configuration in the startup configuration.
show radius-server [*hostname* | *ipv4-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured RADIUS server parameters.

Verifying RADIUS Change of Authorization Configuration

To display RADIUS Change of Authorization configuration information, perform one of the following tasks:

Command	Purpose
show running-config dot1x	Displays the dot1x configuration in the running configuration.
show running-config aaa	Displays the AAA configuration in the running configuration.
show running-config radius	Displays the RADIUS configuration in the running configuration.
show aaa server radius statistics	Displays the local RADIUS server statistics.
show aaa client radius statistics {*ip address* | *hostname* }	Displays the local RADIUS client statistics.
clear aaa server radius statistics	Clears the local RADIUS server statistics.
clear aaa client radius statistics {*ip address* | *hostname* }	Clears the local RADIUS client statistics.

Monitoring RADIUS Servers

You can monitor the statistics that the Cisco NX-OS device maintains for RADIUS server activity.

Before you begin

Configure one or more RADIUS server hosts.

Procedure

	Command or Action	Purpose
Step 1	**show radius-server statistics** {*hostname* | *ipv4-address*}	Displays the RADIUS statistics.

	Command or Action	Purpose
	Example: `switch# show radius-server statistics 10.10.1.1`	

Related Topics

[Configuring RADIUS Server Hosts](), on page 57

[Clearing RADIUS Server Statistics](), on page 76

Clearing RADIUS Server Statistics

You can display the statistics that the Cisco NX-OS device maintains for RADIUS server activity.

Before you begin

Configure RADIUS servers on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	(Optional) **show radius-server statistics** {*hostname* | *ipv4-address*}
	Example:
	`switch# show radius-server statistics 10.10.1.1`
		Displays the RADIUS server statistics on the Cisco NX-OS device.

Step 2	**clear radius-server statistics** {*hostname* | *ipv4-address*}
	Example:
	`switch# clear radius-server statistics 10.10.1.1`
		Clears the RADIUS server statistics.

Related Topics

[Configuring RADIUS Server Hosts](), on page 57

Configuration Example for RADIUS

The following example shows how to configure RADIUS:

```
radius-server key 7 "ToIkLhPpG"
radius-server host 10.10.1.1 key 7 "ShMoMhTl" authentication accounting
aaa group server radius RadServer
    server 10.10.1.1
```

Configuration Examples of RADIUS Change of Authorization

The following example shows how to configure RADIUS Change of Authorization:

```
radius-server host 10.77.143.170 key 7 "fewhg123" authentication accounting
aaa server radius dynamic-author
    client 10.77.143.170 vrf management server-key 7 "fewhg123"
```

Additional References for RADIUS

This section describes additional information related to implementing RADIUS.

Related Documents

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus® 3550-T Unicast Routing Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
MIBs related to RADIUS	To locate and download supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

CHAPTER 7

Configuring IP ACLs

This chapter describes how to configure IP access control lists (ACLs) on Cisco NX-OS devices.

Unless otherwise specified, the term IP ACL refers to IPv4 ACLs.

This chapter includes the following sections:

About ACLs

An ACL is an ordered set of rules that you can use to filter traffic. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the device determines that an ACL applies to a packet, it tests the packet against the conditions of all rules. The first matching rule determines whether the packet is permitted or denied. If there is no match, the device applies the applicable implicit rule. The device continues processing packets that are permitted and drops packets that are denied.

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

ACL Types and Applications

The device supports the following types of ACLs for security traffic filtering:

IPv4 ACLs

The Cisco Nexus® 3550-T device applies IPv4 ACLs only to IPv4 TCP and UDP traffic.

IP has the following types of applications:

Router ACL
Filters Layer 3 traffic

VTY ACL
Filters virtual teletype (VTY) traffic

Note Only Router and VTY ACL IP applications are supported in Cisco Nexus® 3550-T.

Note Only the ingress policy can be configured in Cisco Nexus® 3550-T switches to filter the ingress traffic based on conditions specified in the ACL on the following interfaces:

- Physical Layer 3 interfaces
- Layer 3 Ethernet port-channel interfaces

This table summarizes the applications for security ACLs.

Table 7: Security ACL Applications

Application	Supported Interfaces	Types of ACLs Supported
Router ACL	• Physical Layer 3 interfaces • Layer 3 Ethernet port-channel interfaces • Management interfaces	• IPv4 ACLs **Note** Egress router ACLs are **not supported** on Cisco Nexus® 3550-T switch uplink ports.

Order of ACL Application

When the device processes a packet, it determines the forwarding path of the packet. The path determines which ACLs that the device applies to the traffic. The device only applies the Ingress router ACL.

About Rules

Rules are what you create, modify, and remove when you configure how an ACL filters network traffic. Rules appear in the running configuration. When you apply an ACL to an interface or change a rule within an ACL that is already applied to an interface, the supervisor module creates ACL entries from the rules in the running configuration and sends those ACL entries to the applicable I/O module. Depending upon how you configure the ACL, there may be more ACL entries than rules, especially if you implement policy-based ACLs by using object groups when you configure rules.

You can create rules in access-list configuration mode by using the **permit** or **deny** command. The device allows traffic that matches the criteria in a permit rule and blocks traffic that matches the criteria in a deny rule. You have many options for configuring the criteria that traffic must meet in order to match the rule.

This section describes some of the options that you can use when you configure a rule.

Source and Destination

In each rule, you specify the source and the destination of the traffic that matches the rule. You can specify both the source and destination as a specific host, a network or group of hosts, or any host.

Implicit Rules for IP ACL

IP ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the device applies them to traffic when no other rules in an ACL match.

All IPv4 ACLs include the following implicit rule:

```
deny ip any any
```

This implicit rule ensures that the device denies unmatched IP traffic.

This implicit rule ensures that the device denies the unmatched traffic, regardless of the protocol specified in the Layer 2 header of the traffic.

Additional Filtering Options

You can identify traffic by using additional options. These options differ by ACL type. The following list includes most but not all additional filtering options:

- IPv4 ACLs support the following additional filtering options:
  - Layer 4 protocol
  - TCP and UDP ports
  - ICMP types and codes

Sequence Numbers

The device supports sequence numbers for rules. Every rule that you enter receives a sequence number, either assigned by you or assigned automatically by the device. Sequence numbers simplify the following ACL tasks:

Adding new rules between existing rules

By specifying the sequence number, you specify where in the ACL a new rule should be positioned. For example, if you need to insert a rule between rules numbered 100 and 110, you could assign a sequence number of 105 to the new rule.

Removing a rule

Without using a sequence number, removing a rule requires that you enter the whole rule, as follows:

```
switch(config-acl)# no permit tcp 10.0.0.0/8 any
```

However, if the same rule had a sequence number of 101, removing the rule requires only the following command:

```
switch(config-acl)# no 101
```

Moving a rule

With sequence numbers, if you need to move a rule to a different position within an ACL, you can add a second instance of the rule using the sequence number that positions it correctly, and then you can remove the original instance of the rule. This action allows you to move the rule without disrupting traffic.

If you enter a rule without a sequence number, the device adds the rule to the end of the ACL and assigns a sequence number that is 10 greater than the sequence number of the preceding rule to the rule. For example, if the last rule in an ACL has a sequence number of 225 and you add a rule without a sequence number, the device assigns the sequence number 235 to the new rule.

In addition, Cisco NX-OS allows you to reassign sequence numbers to rules in an ACL. Resequencing is useful when an ACL has rules numbered contiguously, such as 100 and 101, and you need to insert one or more rules between those rules.

Logical Operators and Logical Operation Units

IP ACL rules for TCP and UDP traffic can use logical operators to filter traffic based on port numbers. Cisco NX-OS supports logical operators in only the ingress direction.

The device stores operator-operand couples in registers called logical operator units (LOUs). The LOU usage for each type of operator is as follows:

eq
Is never stored in an LOU

gt
Uses 1 LOU

lt
Uses 1 LOU

range
Uses 1 LOU

Time Ranges

You can use time ranges to control when an ACL rule is in effect. For example, if the device determines that a particular ACL applies to traffic arriving on an interface, and a rule in the ACL uses a time range that is not in effect, the device does not compare the traffic to that rule. The device evaluates time ranges based on its clock.

When you apply an ACL that uses time ranges, the device updates the affected I/O module whenever a time range referenced in the ACL starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.

IPv4 ACLs support time ranges. When the device applies an ACL to traffic, the rules in effect are as follows:

- All rules without a time range specified
- Rules with a time range that includes the second when the device applies the ACL to traffic

The device supports named, reusable time ranges, which allows you to configure a time range once and specify it by name when you configure many ACL rules. Time range names have a maximum length of 64 alphanumeric characters.

A time range contains one or more rules. The two types of rules are as follows:

Absolute

A rule with a specific start date and time, specific end date and time, both, or neither. The following items describe how the presence or absence of a start or end date and time affect whether an absolute time range rule is active:

- Start and end date and time both specified—The time range rule is active when the current time is later than the start date and time and earlier than the end date and time.
- Start date and time specified with no end date and time—The time range rule is active when the current time is later than the start date and time.
- No start date and time with end date and time specified—The time range rule is active when the current time is earlier than the end date and time.
- No start or end date and time specified—The time range rule is always active.

For example, you could prepare your network to allow access to a new subnet by specifying a time range that allows access beginning at midnight of the day that you plan to place the subnet online. You can use that time range in ACL rules that apply to the subnet. After the start time and date have passed, the device automatically begins applying the rules that use this time range when it applies the ACLs that contain the rules.

Periodic

A rule that is active one or more times per week. For example, you could use a periodic time range to allow access to a lab subnet only during work hours on weekdays. The device automatically applies ACL rules that use this time range only when the range is active and when it applies the ACLs that contain the rules.

Note The order of rules in a time range does not affect how a device evaluates whether a time range is active. Cisco NX-OS includes sequence numbers in time ranges to make editing the time range easier.

Time ranges also allow you to include remarks, which you can use to insert comments into a time range. Remarks have a maximum length of 100 alphanumeric characters.

The device determines whether a time range is active as follows:

- The time range contains one or more absolute rules—The time range is active if the current time is within one or more absolute rules.
- The time range contains one or more periodic rules—The time range is active if the current time is within one or more periodic rules.
- The time range contains both absolute and periodic rules—The time range is active if the current time is within one or more absolute rules and within one or more periodic rules.

When a time range contains both absolute and periodic rules, the periodic rules can only be active when at least one absolute rule is active.

Prerequisites for IP ACLs

IP ACLs have the following prerequisites:

- You must be familiar with IP addressing and protocols to configure IP ACLs.
- You must be familiar with the interface types that you want to configure with ACLs.

Guidelines and Limitations for IP ACLs

IP ACLs have the following configuration guidelines and limitations:

- Duplicate ACL entries with different sequence numbers are allowed in the configuration. However, these duplicate entries are not programmed in the hardware access-list.
- Usually, ACL processing for IP packets occurs on the I/O modules, which use hardware that accelerates ACL processing. In some circumstances, processing occurs on the supervisor module, which can result in slower ACL processing, especially during processing that involves an ACL with many rules. Management interface traffic is always processed on the supervisor module. If IP packets in any of the following categories are exiting a Layer 3 interface, they are sent to the supervisor module for processing:
    - IPv4 packets that have IP options (other IP packet header fields following the destination address field).

  In Cisco Nexus® 3550-T switches Storm control settings are used to prevent redirected packets from overwhelming the supervisor module.

  For more information on storm control, see Configuring Traffic Storm Control, on page 47

  .
- When you apply an ACL that uses time ranges, the device updates the ACL entries whenever a time range that is referenced in an ACL entry starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.
- The VTY ACL feature restricts all traffic for all VTY lines. You cannot specify different traffic restrictions for different VTY lines. Any router ACL can be configured as a VTY ACL.
- An egress VTY ACL (an IP ACL applied to the VTY line in the outbound direction) prevents the switch from copying files using a file transfer protocol (TFTP, FTP, SCP, SFTP, etc.) unless the file transfer protocol is explicitly permitted within the egress VTY ACL.
- When you apply an undefined ACL to an interface, the system treats the ACL as empty and permits all traffic.
- IP tunnels do not support ACLs or QoS policies.
- IPv4 ACL logging in the egress direction is not supported.
- ACL logging applies to port ACLs configured by the **ip port access-group** command and to router ACLs configured by the **ip access-group** command only.
- The total number of IPv4 ACL flows is limited to a user-defined maximum value to prevent DoS attacks. If this limit is reached, no new logs are created until an existing flow finishes.
- The number of syslog entries that are generated by IPv4 ACL logging is limited by the configured logging level of the ACL logging process. If the number of syslog entries exceeds this limit, the logging facility might drop some logging messages. Therefore, IPv4 ACL logging should not be used as a billing tool or as an accurate source of the number of matches to an ACL.

- A router ACL applied on a Layer 3 physical or logical interface does not match multicast traffic. This behavior applies to Cisco Nexus® 3550-T switches.
- If the same QoS policy and ACL are applied to multiple interfaces, the label is shared only when the QoS policy is applied with the no-stats option.
- Access-lists based on HTTP methods are not supported on the Cisco Nexus® 3550-T Platform switches and the Cisco Nexus® 3550-T switches.
- The following guidelines and limitations apply to Cisco Nexus® 3550-T switches:
    - RACLs cannot match on packets with multicast MAC destination addresses.

Default Settings for IP ACLs

This table lists the default settings for IP ACL parameters.

Table 8: Default IP ACL Parameters

Parameters	Default
IP ACLs	No IP ACLs exist by default
IP ACL entries	1024
ACL rules	Implicit rules apply to all ACLs
Object groups	No object groups exist by default
Time ranges	No time ranges exist by default

Configuring IP ACLs

Creating an IP ACL

You can create an IPv4 ACL on the device and add rules to it.

Before you begin

This feature allows you to verify the ACL configuration and confirm that the resources that are required by the configuration are available before committing them to the running configuration. This feature is especially useful for ACLs that include more than about 1000 rules.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose

	Example:
	`switch# configure terminal`
	`switch(config)#`
		Note When ACL is enabled only TCP and UDP packets are handled in the Cisco Nexus® 3550-T hardware.

Step 2	Enter the following commands: **ip access-list** *name*
	Example:
	`switch(config)# ip access-list acl-01`
	`switch(config-acl)#`
		Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.

Step 3	[*sequence-number*] {**permit** | **deny**} *protocol* {*source-ip-prefix* | *source-ip-mask*} {*destination-ip-prefix* | *destination-ip-mask*}	Creates a rule in the IP ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For IPv4 access lists, you can specify a source and destination IPv4 prefix, which matches only on the first contiguous bits, or you can specify a source and destination IPv4 wildcard mask, which matches on any bit in the address.
Step 4	(Optional) Enter the following commands: **show ip access-lists** *name* **Example:** `switch(config-acl)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Changing an IP ACL

You can add and remove rules in an existing IPv4 ACL, but you cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers.

Before you begin

This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. This feature is especially useful for ACLs that include more than about 1000 rules.

Procedure

	Command or Action	Purpose

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

Step 2	Enter the following commands: **ip access-list** *name*
	Example:
	`switch(config)# ip access-list acl-01`
	`switch(config-acl)#`
		Enters IP ACL configuration mode for the ACL that you specify by name.

Step 3	(Optional) [*sequence-number*] {**permit** | **deny**} *protocol source destination* **Example:** `switch(config-acl)# 100 permit ip 192.168.2.0/24 any`	Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic.

Step 4	(Optional) **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*}
	Example:
	`switch(config-acl)# no 80`
		Removes the rule that you specified from the IP ACL.
		The **permit** and **deny** commands support many ways of identifying traffic.

Step 5	(Optional) Enter the following commands: **show ip access-lists** *name* **Example:** `switch(config-acl)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Changing Sequence Numbers in an IP ACL

You can change all the sequence numbers assigned to the rules in an IP ACL.

Before you begin

This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. This feature is especially useful for ACLs that include more than about 1000 rules.

Procedure

	Command or Action	Purpose

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

Step 2	**resequence** {**ip** | **ipv4**} **access-list** *name starting-sequence-number increment* **Example:** `switch(config)# resequence access-list ip acl-01 100 10`	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. The *starting-sequence-number* argument and the *increment* argument can be a whole number between 1 and 4294967295.
Step 3	(Optional) **show ip access-lists** *name* **Example:** `switch(config)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Removing an IP ACL

You can remove an IP ACL from the device.

Before you begin

Ensure that you know whether the ACL is applied to an interface. The device allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, the device considers the removed ACL to be empty. Use the **show ip access-lists** command with the summary keyword to find the interfaces that an IP ACL is configured on.

Procedure

	Command or Action	Purpose

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

Step 2	Enter the following commands: **no ip access-list** *name* **Example:** `switch(config)# no ip access-list acl-01`	Removes the IP ACL that you specified by name from the running configuration.
Step 3	(Optional) Enter the following commands: **show ip access-lists** *name* **summary** **Example:** `switch(config)# show ip access-lists acl-01 summary`	Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Applying an IP ACL as a Router ACL

You can apply an IPv4 ACL to any of the following types of interfaces:

- Physical Layer 3 interfaces and subinterfaces
- Layer 3 Ethernet port-channel interfaces

ACLs applied to these interface types are considered router ACLs.

Before you begin

Ensure that the ACL you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 1/3`
	`switch(config-if)#`
		Enters configuration mode for the interface type that you specified.

Step 3	Enter the following commands: **ip access-group** *access-list* **Example:** `switch(config-if)# ip access-group acl1 in`	Applies an IPv4 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 4	(Optional) **show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	Displays the ACL configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the IP ACL Configuration

To display IP ACL configuration information, perform one of the following tasks.

Command	Purpose
show ip access-lists	Displays the IPv4 ACL configuration.
show running-config aclmgr [**all**]	Displays the ACL running configuration, including the IP ACL configuration and the interfaces to which IP ACLs are applied.

Command	Purpose
show startup-config aclmgr [**all**]	Displays the ACL startup configuration. **Note** This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default and user-configured ACLs in the startup configuration.

Configuration Examples for IP ACLs

The following example shows how to create an IPv4 ACL named acl-01 and apply it as a RACL to Ethernet interface 1/1, which is a Layer 3 interface:

```
ip access-list acl-01
  permit ip 192.168.2.0/24 any
interface ethernet 1/1
  ip port access-group acl-01 in
```

Configuring Time-Ranges

Creating a Time-Range

You can create a time range on the device and add rules to it.

Procedure

	Command or Action	Purpose

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

Step 2	**time-range** *name*
	Example:
	`switch(config)# time-range workday-daytime`
	`switch(config-time-range)#`
		Creates the time range and enters time-range configuration mode.

	Command or Action	Purpose
Step 3	(Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time* **Example:** `switch(config-time-range)# periodic monday 00:00:00 to friday 23:59:59`	Creates a periodic rule that is in effect for one or more contiguous days between and including the specified start and end days and times.
Step 4	(Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time* **Example:** `switch(config-time-range)# periodic weekdays 06:00:00 to 20:00:00`	Creates a periodic rule that is in effect on the days specified by the *list-of-weekdays* argument between and including the specified start and end times. The following keywords are also valid values for the *list-of-weekdays* argument: • **daily** —All days of the week. • **weekdays** —Monday through Friday. • **weekend** —Saturday through Sunday.
Step 5	(Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*] **Example:** `switch(config-time-range)# absolute start 1:00 15 march 2013`	Creates an absolute rule that is in effect beginning at the time and date specified after the **start** keyword. If you omit the **end** keyword, the rule is always in effect after the start time and date have passed.
Step 6	(Optional) [*sequence-number*] **absolute** [start *time date*] **end** *time date* **Example:** `switch(config-time-range)# absolute end 23:59:59 31 may 2013`	Creates an absolute rule that is in effect until the time and date specified after the **end** keyword. If you omit the **start** keyword, the rule is always in effect until the end time and date have passed.
Step 7	(Optional) **show time-range** *name* **Example:** `switch(config-time-range)# show time-range workday-daytime`	Displays the time-range configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-time-range)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Changing a Time-Range

You can add and remove rules in an existing time range. You cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers.

Procedure

	Command or Action	Purpose

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

Step 2	**time-range** *name*
	Example:
	`switch(config)# time-range workday-daytime`
	`switch(config-time-range)#`
		Enters time-range configuration mode for the specified time range.

Step 3	(Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time* **Example:** `switch(config-time-range)# periodic monday 00:00:00 to friday 23:59:59`	Creates a periodic rule that is in effect for one or more contiguous days between and including the specified start and end days and times.
Step 4	(Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time* **Example:** `switch(config-time-range)# 100 periodic weekdays 05:00:00 to 22:00:00`	Creates a periodic rule that is in effect on the days specified by the *list-of-weekdays* argument between and including the specified start and end times. The following keywords are also valid values for the *list-of-weekdays* argument: • **daily** —All days of the week. • **weekdays** —Monday through Friday. • **weekend** —Saturday through Sunday.
Step 5	(Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*] **Example:** `switch(config-time-range)# absolute start 1:00 15 march 2013`	Creates an absolute rule that is in effect beginning at the time and date specified after the **start** keyword. If you omit the **end** keyword, the rule is always in effect after the start time and date have passed.
Step 6	(Optional) [*sequence-number*] **absolute** [**start** *time date*] **end** *time date* **Example:** `switch(config-time-range)# absolute end 23:59:59 31 may 2013`	Creates an absolute rule that is in effect until the time and date specified after the **end** keyword. If you omit the **start** keyword, the rule is always in effect until the end time and date have passed.
Step 7	(Optional) **no** {*sequence-number* | **periodic** *arguments . . .* | **absolute** *arguments. . .*}	Removes the specified rule from the time range.

	Command or Action	Purpose
	Example: `switch(config-time-range)# no 80`	
Step 8	(Optional) **show time-range** *name* **Example:** `switch(config-time-range)# show time-range workday-daytime`	Displays the time-range configuration.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-time-range)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Removing a Time-Range

You can remove a time range from the device.

Before you begin

Ensure that you know whether the time range is used in any ACL rules. The device allows you to remove time ranges that are used in ACL rules. Removing a time range that is in use in an ACL rule does not affect the configuration of interfaces where you have applied the ACL. Instead, the device considers the ACL rule using the removed time range to be empty.

Procedure

	Command or Action	Purpose

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

Step 2	**no time-range** *name* **Example:** `switch(config)# no time-range daily-workhours`	Removes the time range that you specified by name.
Step 3	(Optional) **show time-range** **Example:** `switch(config-time-range)# show time-range`	Displays the configuration for all time ranges. The removed time range should not appear.
Step 4	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch# copy running-config startup-config`	

Changing Sequence Numbers in a Time Range

You can change all the sequence numbers assigned to rules in a time range.

Procedure

	Command or Action	Purpose

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

Step 2	**resequence time-range** *name starting-sequence-number increment*
	Example:
	`switch(config)# resequence time-range daily-workhours 100 10`
	`switch(config)#`
		Assigns sequence numbers to the rules contained in the time range, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify.

Step 3	(Optional) **show time-range** *name* **Example:** `switch(config)# show time-range daily-workhours`	Displays the time-range configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the Time-Range Configuration

To display time-range configuration information, perform one of the following tasks.

Command	Purpose
show time-range	Displays the time-range configuration.
show running-config aclmgr	Displays ACL configuration, including all time ranges.

CHAPTER 8

Configuring SSH and Telnet

This chapter describes how to configure Secure Shell Protocol (SSH) and Telnet on Cisco NX-OS devices.

This chapter includes the following sections:

About SSH and Telnet

This section includes information about SSH and Telnet.

SSH Server

You can use the SSH server to enable an SSH client to make a secure, encrypted connection to a Cisco NX-OS device. SSH uses strong encryption for authentication. The SSH server in the Cisco NX-OS software can interoperate with publicly and commercially available SSH clients.

The user authentication mechanisms supported for SSH are RADIUS, TACACS+, LDAP, and the use of locally stored usernames and passwords.

SSH Client

The SSH client feature is an application that runs over the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco NX-OS device to make a secure, encrypted connection to another Cisco NX-OS device or to any other device that runs the SSH server. This connection provides an outbound connection that is encrypted. With authentication and encryption, the SSH client allows for a secure communication over an insecure network.

The SSH client in the Cisco NX-OS software works with publicly and commercially available SSH servers.

SSH Server Keys

SSH requires server keys for secure communications to the Cisco NX-OS device. You can use SSH server keys for the following SSH options:

- SSH version 2 using Rivest, Shamir, and Adelman (RSA) public-key cryptography
- SSH version 2 using the Digital System Algrorithm (DSA)

Be sure to have an SSH server key-pair with the appropriate version before enabling the SSH service. You can generate the SSH server key-pair according to the SSH client version used. The SSH service accepts the following types of key-pairs for use by SSH version 2:

- The **dsa** option generates the DSA key-pair for the SSH version 2 protocol.
- The **rsa** option generates the RSA key-pair for the SSH version 2 protocol.

By default, the Cisco NX-OS software generates an RSA key using 1024 bits.

SSH supports the following public key formats:

- OpenSSH
- IETF Secure Shell (SECSH)
- Public Key Certificate in Privacy-Enhanced Mail (PEM)

Caution If you delete all of the SSH keys, you cannot start the SSH services.

SSH Authentication Using Digital Certificates

SSH authentication on Cisco NX-OS devices provide X.509 digital certificate support for host authentication. An X.509 digital certificate is a data item that ensures the origin and integrity of a message. It contains encryption keys for secured communications and is signed by a trusted certification authority (CA) to verify the identity of the presenter. The X.509 digital certificate support provides either DSA or RSA algorithms for authentication.

The certificate infrastructure uses the first certificate that supports the Secure Socket Layer (SSL) and is returned by the security infrastructure, either through a query or a notification. Verification of certificates is successful if the certificates are from any of the trusted CAs configured and if not revoked or expired.

You can configure your device for SSH authentication using an X.509 certificate. If the authentication fails, you are prompted for a password.

Telnet Server

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site and then passes the keystrokes from one device to the other. Telnet can accept either an IP address or a domain name as the remote device address.

The Telnet server is disabled by default on the Cisco NX-OS device.

Prerequisites for SSH and Telnet

Make sure that you have configured IP on a Layer 3 interface, out-of-band on the mgmt 0 interface, or inband on an Ethernet interface.

Guidelines and Limitations for SSH and Telnet

SSH and Telnet have the following configuration guidelines and limitations:

- The Cisco NX-OS software supports only SSH version 2 (SSHv2).
- When you use the **no feature ssh feature** command, port 22 is not disabled . Port 22 is always open and a deny rule is pushed to deny all incoming external connections.
- Due to a Poodle vulnerability, SSLv3 is no longer supported.
- IPSG is not supported on the following:
    - The last six 40-Gb physical ports on the Cisco Nexus® 3550-T switches.
    - All 40G physical ports on the Cisco Nexus® 3550-T switches.
- You can configure your device for SSH authentication using an X.509 certificate. If the authentication fails, you are prompted for a password.
- The SFTP server feature does not support the regular SFTP **chown** and **chgrp** commands.
- When the SFTP server is enabled, only the admin user can use SFTP to access the device.
- SSH public and private keys imported into user accounts that are remotely authenticated through a AAA protocol (such as RADIUS or TACACS+) for the purpose of SSH Passwordless File Copy will not persist when the Nexus device is reloaded unless a local user account with the same name as the remote user account is configured on the device before the SSH keys are imported.
- SSH timeout period must be longer than the time of the tac-pac generation time. Otherwise, the VSH log might show %VSHD-2-VSHD_SYSLOG_EOL_ERR error. Ideally, set to 0 (infinity) before collecting tac-pac or showtech.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings for SSH and Telnet

This table lists the default settings for SSH and Telnet parameters.

Table 9: Default SSH and Telnet Parameters

Parameters	Default
SSH server	Enabled
SSH server key	RSA key generated with 1024 bits
RSA key bits for generation	1024
Telnet server	Disabled
Telnet port number	23
Maximum number of SSH login attempts	3
SCP server	Disabled
SFTP server	Disabled

Configuring SSH

This section describes how to configure SSH.

Generating SSH Server Keys

You can generate an SSH server key based on your security requirements. The default SSH server key is an RSA key that is generated using 1024 bits.

Procedure

	Command or Action	Purpose

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

Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.
Step 3	**feature ssh** **Example:**	Enables SSH.

	Command or Action	Purpose
	`switch(config)# feature ssh`	

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 5	(Optional) **show ssh key** [**dsa** | **rsa** |] [] **Example:** `switch# show ssh key`	Displays the SSH server keys.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Specifying the SSH Public Keys for User Accounts

You can configure an SSH public key to log in using an SSH client without being prompted for a password. You can specify the SSH public key in one of these formats:

- OpenSSH format
- IETF SECSH format

Specifying the SSH Public Keys in IETF SECSH Format

You can specify the SSH public keys in IETF SECSH format for user accounts.

Before you begin

Generate an SSH public key in IETF SCHSH format.

Procedure

	Command or Action	Purpose
Step 1	**copy** *server-file* **bootflash:***filename* **Example:** `switch# copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub`	Downloads the file containing the SSH key in IETF SECSH format from a server. The server can be FTP, secure copy (SCP), secure FTP (SFTP), or TFTP.

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

	Command or Action	Purpose
Step 3	**username** *username* **sshkey file bootflash:***filename* **Example:** `switch(config)# username User1 sshkey file bootflash:secsh_file.pub`	Configures the SSH public key in IETF SECSH format.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 5	(Optional) **show user-account** **Example:** `switch# show user-account`	Displays the user account configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Specifying the SSH Public Keys in OpenSSH Format

You can specify the SSH public keys in OpenSSH format for user accounts.

Before you begin

Generate an SSH public key in OpenSSH format.

Procedure

	Command or Action	Purpose

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

Step 2	**username** *username* **sshkey** *ssh-key*
	Example:
	`switch(config)# username User1 sshkey ssh-rsa`
	`AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50rv7gsEPj`
	`hOBYmsi6PAVKui1nIf/DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQW3g9igG30c6k6+`
	`XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH3UD/vKyziEh5S4Tplx8=`
		Configures the SSH public key in OpenSSH format.

	Command or Action	Purpose

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show user-account** **Example:** `switch# show user-account`	Displays the user account configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a Maximum Number of SSH Login Attempts

You can configure the maximum number of SSH login attempts. If the user exceeds the maximum number of permitted attempts, the session disconnects.

Note The total number of login attempts includes attempts through public-key authentication, certificate-based authentication, and password-based authentication. If public-key authentication is enabled, it takes priority. If only certificate-based and password-based authentication are enabled, certificate-based authentication takes priority. If you exceed the configured number of login attempts through all of these methods, a message appears indicating that too many authentication failures have occurred.

Procedure

	Command or Action	Purpose

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

Step 2	**ssh login-attempts** *number* **Example:** `switch(config)# ssh login-attempts 5`	Configures the maximum number of times that a user can attempt to log into an SSH session. The default maximum number of login attempts is 3. The range is from 1 to 10.

	Command or Action	Purpose
		Note The **no** form of this command removes the previous login attempts value and sets the maximum number of login attempts to the default value of 3.
Step 3	(Optional) **show running-config security all** **Example:** `switch(config)# show running-config security all`	Displays the configured maximum number of SSH login attempts.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Starting SSH Sessions

You can start SSH sessions using IPv4 to connect to remote devices from the Cisco NX-OS device.

Before you begin

Obtain the hostname for the remote device and, if needed, the username on the remote device.

Enable the SSH server on the remote device.

Procedure

	Command or Action	Purpose

Step 1	**ssh** [*username@*]{*ipv4-address* | *hostname*}
	Example:
	`switch# ssh 10.10.1.1`
		Creates an SSH IPv4 session to a remote device using IPv4.

Starting SSH Sessions from Boot Mode

You can start SSH sessions from the boot mode of the Cisco NX-OS device to connect to remote devices.

Before you begin

Obtain the hostname for the remote device and, if needed, the username on the remote device.

Enable the SSH server on the remote device.

Procedure

	Command or Action	Purpose
Step 1	**ssh** [*username***@**]*hostname* **Example:** `switch(boot)# ssh user1@10.10.1.1`	Creates an SSH session to a remote device from the boot mode of the Cisco NX-OS device.
Step 2	**exit** **Example:** `switch(boot)# exit`	Exits boot mode.
Step 3	**copy scp:**//[*username***@**]*hostname*/*filepath directory* **Example:** `switch# copy scp://user1@10.10.1.1/users abc`	Copies a file from the Cisco NX-OS device to a remote device using the Secure Copy Protocol (SCP).

Configuring SSH Passwordless File Copy

You can copy files from a Cisco NX-OS device to a secure copy (SCP) or secure FTP (SFTP) server without a password. To do so, you must create an RSA or DSA identity that consists of public and private keys for authentication with SSH.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **username** *username* **keypair generate** {**rsa** [*bits* [**force**]] | **dsa** [**force**]}
	Example:
	`switch(config)# username user1 keypair generate rsa 2048 force`
		Generates the SSH public and private keys and stores them in the home directory ($HOME/.ssh) of the Cisco NX-OS device for the specified user. The Cisco NX-OS device uses the keys to communicate with the SSH server on the remote machine.
		The *bits* argument is the number of bits used to generate the key. The range is from 768 to 2048. The default value is 1024.
		Use the **force** keyword to replace an existing key. The SSH keys are not generated if the **force** keyword is omitted and SSH keys are already present.

Step 3	(Optional) **show username** *username* **keypair**	Displays the public key for the specified user.

	Command or Action	Purpose
	Example: `switch(config)# show username user1 keypair`	**Note** For security reasons, this command does not show the private key.
Step 4	Required: **username** *username* **keypair export** {**bootflash:***filename* | **volatile:***filename*} {**rsa** | **dsa**} [**force**] **Example:** `switch(config)# username user1 keypair export bootflash:key_rsa rsa`	Exports the public and private keys from the home directory of the Cisco NX-OS device to the specified bootflash or volatile directory. Use the **force** keyword to replace an existing key. The SSH keys are not exported if the **force** keyword is omitted and SSH keys are already present. To export the generated key pair, you are prompted to enter a passphrase that encrypts the private key. The private key is exported as the file that you specify, and the public key is exported with the same filename followed by a .pub extension. You can now copy this key pair to any Cisco NX-OS device and use SCP or SFTP to copy the public key file (*.pub) to the home directory of the server. **Note** For security reasons, this command can be executed only from global configuration mode.
Step 5	Required: **username** *username* **keypair import** {**bootflash:***filename* | **volatile:***filename*} {**rsa** | **dsa**} [**force**] **Example:** `switch(config)# username user1 keypair import bootflash:key_rsa rsa`	Imports the exported public and private keys from the specified bootflash or volatile directory to the home directory of the Cisco NX-OS device. Use the **force** keyword to replace an existing key. The SSH keys are not imported if the **force** keyword is omitted and SSH keys are already present. To import the generated key pair, you are prompted to enter a passphrase that decrypts the private key. The private key is imported as the file that you specify, and the public key is imported with the same filename followed by a .pub extension. **Note** For security reasons, this command can be executed only from global configuration mode. **Note** Only the users whose keys are configured on the server are able to access the server without a password.

What to do next

On the SCP or SFTP server, use the following command to append the public key stored in the *.pub file (for example, key_rsa.pub) to the authorized_keys file:

$ cat key_rsa.pub >> $HOME/.ssh/ authorized_keys

You can now copy files from the Cisco NX-OS device to the server without a password using standard SSH and SCP commands.

Configuring SCP and SFTP Servers

You can configure an SCP or SFTP server on the Cisco NX-OS device in order to copy files to and from a remote device. After you enable the SCP or SFTP server, you can execute an SCP or SFTP command on the remote device to copy the files to or from the Cisco NX-OS device.

Note The arcfour and blowfish cipher options are not supported for the SCP server.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **feature scp-server** **Example:** `switch(config)# feature scp-server`	Enables or disables the SCP server on the Cisco NX-OS device.
Step 3	Required: [**no**] **feature sftp-server** **Example:** `switch(config)# feature sftp-server`	Enables or disables the SFTP server on the Cisco NX-OS device.

Step 4	Required: **exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 5	(Optional) **show running-config security** **Example:** `switch# show running-config security`	Displays the configuration status of the SCP and SFTP servers.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring X.509v3 Certificate-Based SSH Authentication

You can configure SSH authentication using X.509v3 certificates.

Before you begin

Enable the SSH server on the remote device.

Procedure

	Command or Action	Purpose

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

Step 2	**username** *user-id* [**password** [**0** | **5**] *password*] **Example:** `switch(config)# username jsmith password 4Ty18Rnt`	Configures a user account. The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 28 characters. Valid characters are uppercase letters A through Z, lowercase letters a through z, numbers 0 through 9, hyphen (-), period (.), underscore (_), plus sign (+), and equal sign (=). The at symbol (@) is supported in remote usernames but not in local usernames. Usernames must begin with an alphanumeric character. The default password is undefined. The **0** option indicates that the password is clear text, and the **5** option indicates that the password is encrypted. The default is **0** (clear text). **Note** If you do not specify a password, the user might not be able to log in to the Cisco NX-OS device. **Note** If you create a user account with the encrypted password option, the corresponding SNMP user will not be created.

Step 3	**username** *user-id* **ssh-cert-dn** *dn-name* {**dsa** | **rsa**}
	Example:
	`switch(config)# username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1, emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith" rsa`
		Specifies an SSH X.509 certificate distinguished name and DSA or RSA algorithm to use for authentication for an existing user account. The distinguished name can be up to 512 characters and must follow the format shown in the examples. Make sure the email address and state are configured as emailAddress and ST, respectively.

Step 4	[**no**] **crypto ca trustpoint** *trustpoint*	Configures a trustpoint.

	Command or Action	Purpose

	Example:
	`switch(config)# crypto ca trustpoint winca`
	`switch(config-trustpoint)#`
		Note Before you delete a trustpoint using the **no** form of this command, you must first delete the CRL and CA certificate, using the **delete crl** and **delete ca-certificate** commands.

Step 5	**crypto ca authenticate** *trustpoint* **Example:** `switch(config-trustpoint)# crypto ca authenticate winca`	Configures a CA certificate for the trustpoint. **Note** To delete a CA certificate, enter the **delete ca-certificate** command in the trustpoint configuration mode.
Step 6	(Optional) **crypto ca crl request** *trustpoint* **bootflash:***static-crl***.crl** **Example:** `switch(config-trustpoint)# crypto ca crl request winca bootflash:crllist.crl`	This command is optional but highly recommended. Configures the certificate revocation list (CRL) for the trustpoint. The CRL file is a snapshot of the list of revoked certificates by the trustpoint. This static CRL list is manually copied to the device from the Certification Authority (CA). **Note** Static CRL is the only supported revocation check method. **Note** To delete the CRL, enter the **delete crl** command.
Step 7	(Optional) **show crypto ca certificates** **Example:** `switch(config-trustpoint)# show crypto ca certificates`	Displays the configured certificate chain and associated trustpoint.
Step 8	(Optional) **show crypto ca crl** *trustpoint* **Example:** `switch(config-trustpoint)# show crypto ca crl winca`	Displays the contents of the CRL list of the specified trustpoint.
Step 9	(Optional) **show user-account** **Example:** `switch(config-trustpoint)# show user-account`	Displays configured user account details.
Step 10	(Optional) **show users** **Example:** `switch(config-trustpoint)# show users`	Displays the users logged into the device.
Step 11	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config-trustpoint)# copy running-config startup-config`	

Configuring Legacy SSH Algorithm Support

You can configure support for legacy SSH security algorithms, message authentication codes (MACs), key types, and ciphers.

Procedure

	Command or Action	Purpose

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

Step 2	(Optional) **ssh kexalgos** [**all**] **Example:** `switch(config)# ssh kexalgos all`	Use the **all** keyword to enable all supported KexAlgorithms which are the key exchange methods that are used to generate per-connection keys. Supported KexAlgorithmns are: • curve25519-sha256 • diffie-hellman-group-exchange-sha256 • diffie-hellman-group1-sha1 • diffie-hellman-group14-sha1 • diffie-hellman-group1-sha1 • ecdh-sha2-nistp256 • ecdh-sha2-nistp384
Step 3	(Optional) **ssh macs all** **Example:** `switch(config)# ssh macs all`	Enables all supported MACs which are the message authentication codes used to detect traffic modification. Supported MACs are: • hmac-sha1
Step 4	(Optional) **ssh ciphers** [**all**] **Example:** `switch(config)# ssh ciphers all`	Use the **all** keyword to enable all supported ciphers to encrypt the connection. Supported ciphers are: • aes128-cbc • aes192-cbc

	Command or Action	Purpose
		• aes256-cbc • aes128-ctr • aes192-ctr • aes256-ctr • aes256-gcm@openssh.com • aes128-gcm@openssh.com
Step 5	(Optional) **ssh keytypes all** **Example:** `switch(config)# ssh keytypes all`	Enables all supported PubkeyAcceptedKeyTypes which are the public key algorithms that the server can use to authenticate itself to the client. Supported key types are: • ssh-dss • ssh-rsa

Algorithms Supported - FIPs Mode Enabled

The list of algorithms supported when the FIPs mode is enabled are as follows:

Table 10: Algorithms Supported - FIPs Mode Enabled

Algorithms	Supported	Unsupported
ciphers	• aes128-ctr • aes256-ctr • aes256-gcm@openssh.com • aes128-gcm@openssh.com	• aes192-ctr • aes128-cbc • aes192-cbc • aes256-cbc
hmac	• hmac-sha2-256 • hmac-sha2-512 • hmac-sha1	• hmac-sha2-256-etm@openssh.com • hmac-sha2-512-etm@openssh.com • hmac-sha1-etm@openssh.com

Algorithms	Supported	Unsupported
kexalgo	• ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521 • diffie-hellman-group16-sha512 • diffie-hellman-group14-sha1 • diffie-hellman-group14-sha256	• curve25519-sha256 • curve25519-sha256@libssh.org
keytypes	• rsa-sha2-256 • ecdsa-sha2-nistp256 • ecdsa-sha2-nistp384 • ecdsa-sha2-nistp521	ssh-rsa

Changing the Default SSH Server Port

You can change the SSHv2 port number from the default port number 22. Encryptions used while changing the default SSH port provides you with connections that support stronger privacy and session integrity

Procedure

	Command or Action	Purpose

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

Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.

Step 3	**show sockets** *local-port-range*
	Example:
	`switch(config)# show sockets local port range (15001 - 58000)`
	`switch(config)# local port range (58001 - 63535) and nat port range (63536 - 65535)`
		Displays the available port range.

Step 4	**ssh port** *local-port* **Example:** `switch(config)# ssh port 58003`	Configures the port.

	Command or Action	Purpose
Step 5	**feature ssh** **Example:** `switch(config)# feature ssh`	Enables SSH.

Step 6	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 7	(Optional) **show running-config security all** **Example:** `switch# ssh port 58003`	Displays the security configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Clearing SSH Hosts

When you download a file from a server using SCP or SFTP, or when you start an SSH session from this device to a remote host, you establish a trusted SSH relationship with that server. You can clear the list of trusted SSH servers for your user account.

Procedure

	Command or Action	Purpose
Step 1	**clear ssh hosts** **Example:** `switch# clear ssh hosts`	Clears the SSH host sessions and the known host file.

Disabling the SSH Server

By default, the SSH server is enabled on the Cisco NX-OS device. You can disable the SSH server to prevent SSH access to the switch.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show ssh server** **Example:** `switch# show ssh server`	Displays the SSH server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Deleting SSH Server Keys

You can delete SSH server keys on the Cisco NX-OS device after you disable the SSH server.

Note To reenable SSH, you must first generate an SSH server key.

Procedure

	Command or Action	Purpose

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

Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

	Command or Action	Purpose
Step 4	(Optional) **show ssh key** **Example:** `switch# show ssh key`	Displays the SSH server key configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Generating SSH Server Keys, on page 100

Clearing SSH Sessions

You can clear SSH sessions from the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	Displays user session information.
Step 2	**clear line** *vty-line* **Example:** `switch(config)# clear line pts/12`	Clears a user SSH session.

Configuring Telnet

This section describes how to configure Telnet on the Cisco NX-OS device.

Enabling the Telnet Server

You can enable the Telnet server on the Cisco NX-OS device. By default, the Telnet server is disabled.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**feature telnet** **Example:** `switch(config)# feature telnet`	Enables the Telnet server. The default is disabled.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show telnet server** **Example:** `switch# show telnet server`	Displays the Telnet server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Starting Telnet Sessions to Remote Devices

You can start Telnet sessions to connect to remote devices from the Cisco NX-OS device. You can start Telnet sessions using either IPv4.

Before you begin

Obtain the hostname or IP address for the remote device and, if needed, the username on the remote device.

Enable the Telnet server on the Cisco NX-OS device.

Enable the Telnet server on the remote device.

Procedure

	Command or Action	Purpose
Step 1	**telnet** {*ipv4-address* | *host-name*} [*port-number*] **Example:** `switch# telnet 10.10.1.1`	Starts a Telnet session to a remote device using IPv4. The default port number is 23. The range is from 1 to 65535.

Related Topics

Enabling the Telnet Server, on page 115

Clearing Telnet Sessions

You can clear Telnet sessions from the Cisco NX-OS device.

Before you begin

Enable the Telnet server on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	Displays user session information.
Step 2	**clear line** *vty-line* **Example:** `switch(config)# clear line pts/12`	Clears a user Telnet session.

Verifying the SSH and Telnet Configuration

To display the SSH and Telnet configuration information, perform one of the following tasks:

Command	Purpose
show ssh key [**dsa** | **rsa**] []	Displays the SSH server keys.
show running-config security [**all**]	Displays the SSH and user account configuration in the running configuration. The **all** keyword displays the default values for the SSH and user accounts.
show ssh server	Displays the SSH server configuration.
show telnet server	Displays the Telnet server configuration.
show username *username* **keypair**	Displays the public key for the specified user.
show user-account	Displays configured user account details.
show users	Displays the users logged into the device.

Configuration Example for SSH

The following example shows how to configure SSH with an OpenSSH key:

Procedure

Step 1 Disable the SSH server.

Example:

```
switch# configure terminal
switch(config)# no feature ssh
```

Step 2 Generate an SSH server key.

Example:

```
switch(config)# ssh key rsa
generating rsa key(1024 bits)......
generated rsa key
```

Step 3 Enable the SSH server.

Example:

```
switch(config)# feature ssh
```

Step 4 Display the SSH server key.

Example:

Step 5 Specify the SSH public key in OpenSSH format.

Example:

```
switch(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50r
v7gsEPjhOBYmsi6PAVKui1nIf/DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQ
W3g9igG30c6k6+XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH3UD/vKyziEh5
4Tplx8=
```

Step 6 Save the configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Configuration Example for SSH Passwordless File Copy

The following example shows how to copy files from a Cisco NX-OS device to a secure copy (SCP) or secure FTP (SFTP) server without a password:

Procedure

Step 1 Generate the SSH public and private keys and store them in the home directory of the Cisco NX-OS device for the specified user.

Example:

```
switch# configure terminal
switch(config)# username admin keypair generate rsa
generating rsa key(1024 bits)......
```

```
generated rsa key
```

Step 2 Display the public key for the specified user.

Example:

```
switch(config)# show username admin keypair

**************************************

rsa Keys generated: Thu Jul  9 11:10:29 2013

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD0P8boZElTfJ
Fx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvqsrU9TBypYDPQkR/+Y6cKubyFW
VxSBG/NHztQc3+QC1zdkIxGNJbEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbq
S33GZsCAX6v0=

bitcount:262144
fingerprint:
8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d
**************************************

could not retrieve dsa key information
**************************************
```

Step 3 Export the public and private keys from the home directory of the Cisco NX-OS device to the specified bootflash directory.

Example:

```
switch(config)# username admin keypair export bootflash:key_rsa rsa
Enter Passphrase:
switch(config)# dir
.
.
.
        951     Jul 09 11:13:59 2013  key_rsa
        221     Jul 09 11:14:00 2013  key_rsa.pub
.
.
```

Step 4 After copying these two files to another Cisco NX-OS device using the **copy scp** or **copy sftp** command, import them to the home directory of the Cisco NX-OS device.

Example:

```
switch(config)# username admin keypair import bootflash:key_rsa rsa
Enter Passphrase:
switch(config)# show username admin keypair
**************************************

rsa Keys generated: Thu Jul  9 11:10:29 2013

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD0P8boZElTfJ
Fx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvqsrU9TBypYDPQkR/+Y6cKubyFW
VxSBG/NHztQc3+QC1zdkIxGNJbEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbq
S33GZsCAX6v0=
bitcount:262144
fingerprint:
8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d
**************************************

could not retrieve dsa key information
**************************************
switch(config)#
```

Step 5 On the SCP or SFTP server, append the public key stored in key_rsa.pub to the authorized_keys file.

Example:

```
$ cat key_rsa.pub >> $HOME/.ssh/ authorized_keys
```

You can now copy files from the Cisco NX-OS device to the server without a password using standard SSH and SCP commands.

Step 6 (Optional) Repeat this procedure for the DSA keys.

Configuration Example for X.509v3 Certificate-Based SSH Authentication

The following example shows how to configure SSH authentication using X.509v3 certificates:

```
configure terminal
username jsmith password 4Ty18Rnt
username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1,
emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith"
rsa
crypto ca trustpoint tp1
crypto ca authenticate tp1
crypto ca crl request tp1 bootflash:crl1.crl

show crypto ca certificates
Trustpoint: tp1
CA certificate 0:
subject= /CN=SecDevCA
issuer= /CN=SecDevCA
serial=01AB02CD03EF04GH05IJ06KL07MN
notBefore=Jun 29 12:36:26 2016 GMT
notAfter=Jun 29 12:46:23 2021 GMT
SHA1 Fingerprint=47:29:E3:00:C1:C1:47:F2:56:8B:AC:B2:1C:64:48:FC:F4:8D:53:AF
purposes: sslserver sslclient

show crypto ca crl tp1
Trustpoint: tp1 CRL: Certificate Revocation List (CRL):
    Version 2 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: /CN=SecDevCA
    Last Update: Aug 8 20:03:15 2016 GMT
    Next Update: Aug 16 08:23:15 2016 GMT
    CRL extensions:
        X509v3 Authority Key Identifier:
            keyid:30:43:AA:80:10:FE:72:00:DE:2F:A2:17:E4:61:61:44:CE:78:FF:2A
```

```
show user-account
user:user1
        this user account has no expiry date
        roles:network-operator
        ssh cert DN : /C = US, ST = New York, L = Metropolis, O = cisco , OU = csg, CN =
user1; Algo: x509v3-sign-rsa

show users
NAME      LINE        TIME          IDLE      PID         COMMENT
user1     pts/1       Jul 27 18:43  00:03     18796      (10.10.10.1)   session=ssh
```

Additional References for SSH and Telnet

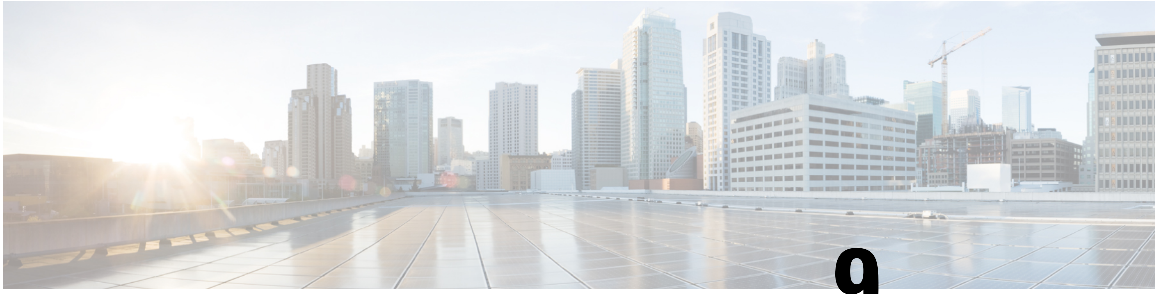
This section describes additional information related to implementing SSH and Telnet.

Related Documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus® 3550-T Unicast Routing Configuration Guide*

MIBs

MIBs	MIBs Link
MIBs related to SSH and Telnet	To locate and download supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 9

Configuring DHCP

This chapter describes how to configure the Dynamic Host Configuration Protocol (DHCP) on a Cisco NX-OS device.

This chapter includes the following sections:

About DHCP Client

The DHCP client feature enables the configuration of an IPv4 address on an interface. Interfaces can include the management port and switch virtual interfaces (SVIs).

Guidelines and Limitations for DHCP

DHCP has the following configuration guidelines and limitations:

- For secure POAP, make sure that DHCP snooping is enabled and firewall rules are set to block unintended or malicious DHCP servers.

Note The firewall rules should be correctly setup for configuring secure POAP.

Note For DHCP configuration limits, see the *Cisco Nexus 3550-T Verified Scalability Guide*.

Configuring DHCP

Enabling DHCP Client

You can use the DHCP client feature to enable the configuration of an IPv4 address on management interface.

Note DHCP client is independent of the DHCP relay processes, so it does not require that the **feature dhcp** command be enabled.

Procedure

	Command or Action	Purpose

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

Step 2	Choose:
	• **interface mgmt 0**
	Example:
	`switch(config)# interface mgmt. 0`
	`switch(config-if)#`
		• Enters interface configuration mode and specifies the management interface as the interface for which you want to enable the DHCP client feature.

Step 3	[**no**] {**ip** } **address dhcp** **Example:** `switch(config-if)# ip address dhcp`	Assigns an IPv4 address to the interface. The **no** form of this command releases the IP address.
Step 4	(Optional) Do the following: • **show running-config interface mgmt 0** **Example:** `switch(config-if)# show running-config interface mgmt. 0`	Displays the IPv4 address assigned to the interface in the running configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration. Only the {**ip**} **address dhcp** command is saved. The assigned IP address is not saved even though it shows in the running configuration.

Configuration Examples for DHCP Client

The following example shows how the DHCP client feature can be used to assign an IPv4 address to a VLAN interface:

```
switch# configure terminal
switch(config)# interface mgmt 0
switch(config-if)# no shutdown
switch(config-if)# ip address dhcp
switch(config-if)# show running-config interface vlan 7
interface Vlan7
no shutdown
ip address dhcp
```

Additional References for DHCP

Related Documents

Related Topic	Document Title
Layer 3 virtualization	*Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*

Standards

PART III

Cisco Nexus 3550-T System Management Configuration Guide

CHAPTER 10

System Management Overview

Software Image

The Cisco NX-OS software consists of one NXOS software image. This image runs on all Cisco Nexus 3550-T switches.

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Cisco Discovery Protocol

You can use the Cisco Discovery Protocol (CDP) to discover and view information about all Cisco equipment that is directly attached to your device. CDP runs on all Cisco-manufactured equipment including routers, bridges, access and communication servers, and switches. CDP is media and protocol independent, and gathers the protocol addresses of neighboring devices, discovering the platform of those devices. CDP runs over the data link layer only. Two systems that support different Layer 3 protocols can learn about each other.

LLDP

Link Layer Discovery Protocol (LLDP) is a vendor-neutral, one-way device discovery protocol that allows network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other. You can enable LLDP globally or per interface.

CHAPTER 11

Configuring CDP

This chapter describes how to configure the Cisco Discovery Protocol (CDP) on Cisco NX-OS devices.

This chapter includes the following sections:

About CDP

The Cisco Discovery Protocol (CDP) is a media-independent and protocol-independent protocol that runs on all Cisco-manufactured equipment including routers, bridges, access and communication servers, and switches. You can use CDP to discover and view information about all the Cisco devices that are directly attached to the device.

CDP gathers protocol addresses of neighboring devices and discovers the platform of those devices. CDP runs over the data link layer only. Two systems that support different Layer 3 protocols can learn about each other.

Each device that you configure for CDP sends periodic advertisements to a multicast address. Each device advertises at least one address at which it can receive SNMP messages. The advertisements also contain hold-time information, which indicates the length of time that a receiving device should hold CDP information before removing it. You can configure the advertisement or refresh timer and the hold timer.

CDP Version-2 (CDPv2) allows you to track instances where the native VLAN ID or port duplex states do not match between connecting devices.

CDP advertises the following type-length-value fields (TLVs):

- Device ID
- Address
- Port ID
- Capabilities
- Version

- Platform
- Native VLAN
- Full or Half Duplex
- SysName
- SysObjectID
- Management Address
- Physical Location

All CDP packets include a VLAN ID. If you configure CDP on a Layer 2 access port, the CDP packets sent from that access port include the access port VLAN ID. If you configure CDP on a Layer 2 trunk port, the CDP packets sent from that trunk port include the lowest configured VLAN ID allowed on that trunk port. The trunk port can receive CDP packets that include any VLAN ID in the allowed VLAN list for that trunk port. For more information on VLANs, see the *Cisco Nexus® 3550-T Layer 2 Switching Configuration* section.

High Availability

Cisco NX-OS supports both stateful and stateless restarts for CDP.

Guidelines and Limitations for CDP

CDP has the following configuration guidelines and limitations:

- CDP can discover up to 256 neighbors per port if the port is connected to a hub with 256 connections.
- CDP must be enabled on the device or you cannot enable it on any interfaces.
- You can configure CDP on physical interfaces and port channels only.

Default Settings for CDP

This table lists the default settings for CDP parameters.

Parameters	Default
CDP	Enabled globally and on all interfaces
CDP version	Version 2
CDP device ID	Serial number
CDP timer	60 seconds
CDP hold timer	180 seconds

Configuring CDP

Note The Cisco NX-OS commands for this feature may differ from those commands that are used in Cisco IOS.

Enabling or Disabling CDP Globally

CDP is enabled by default. You can disable CDP and then reenable it.

You must enable CDP on the device before you enable CDP on any interfaces. If CDP is disabled globally and you enable CDP on specified interfaces, CDP will not be active on those interfaces; the system does not return an error message.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **cdp enable** **Example:** `switch(config)# cdp enable`	Enables or disables the CDP feature on the entire device. It is enabled by default.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling CDP on an Interface

CDP is enabled by default on an interface. You can disable CDP on an interface.

If CDP is disabled globally and you enable CDP on specified interfaces, CDP will not be active on those interfaces; the system does not return an error message.

Procedure

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

	Command or Action	Purpose

	`switch# configure terminal`
	`switch(config)#`	

Step 2	**interface** *interface slot*/*port*
	Example:
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	[**no**] **cdp enable** **Example:** `switch(config-if)# cdp enable`	Enables or disables CDP on this interface. It is enabled by default. **Note** Make sure that CDP is enabled globally on the device.
Step 4	(Optional) **show cdp interface** *interface slot*/*port* **Example:** `switch(config-if)# show cdp interface ethernet 1/2`	Displays CDP information for an interface.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Optional CDP Parameters

You can use the optional commands in this procedure to modify CDP.

Procedure

	Command or Action	Purpose

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

Step 2	(Optional) **cdp advertise** {**v1** | **v2**}
	Example:
	`switch(config)# cdp advertise v1`
		Sets the CDP version that is supported by the device. The default is v2.

Step 3	(Optional) **cdp format device-id** {**mac-address** | **serial-number** | **system-name**}
	Example:
	`switch(config)# cdp format device-id mac-address`
		Sets the CDP device ID. The options are as follows:
		• **mac-address**—The MAC address of the chassis.

	Command or Action	Purpose
		- **serial-number**—The chassis serial number/Organizationally Unique Identifier (OUI). - **system-name**—The system name or fully qualified domain name. The default is **system-name**.
Step 4	(Optional) **cdp holdtime** *seconds* **Example:** `switch(config)# cdp holdtime 150`	Sets the time that CDP holds onto neighbor information before removing it. The range is from 10 to 255 seconds. The default is 180 seconds.
Step 5	(Optional) **cdp timer** *seconds* **Example:** `switch(config)# cdp timer 50`	Sets the refresh time when CDP sends advertisements to neighbors. The range is from 5 to 254 seconds. The default is 60 seconds.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the CDP Configuration

To display the CDP configuration, perform one of the following tasks:

Command	Purpose
show cdp all	Displays all interfaces that have CDP enabled.
show cdp entry {**all** | **name** *entry-name*}	Displays the CDP database entries.
show cdp global	Displays the CDP global parameters.
show cdp interface *interface slot*/*port*	Displays the CDP interface status.
show cdp neighbors {**device-id** | **interface** *interface slot*/*port*} [**detail**]	Displays the CDP neighbor status.
show cdp interface *interface slot*/*port*	Displays the CDP traffic statistics on an interface.

Use the **clear cdp counters** command to clear CDP statistics on an interface.

Use the **clear cdp table** command to clear the CDP cache for one or all interfaces.

It is recommended to use the **show cdp neighbors detail** command instead of **show cdp neighbors** command. The **show cdp neighbors** command can display only 13 characters of a platform name. To get the full platform name in the display, use **show cdp neighbors detail** command.

Configuration Example for CDP

This example shows how to enable the CDP feature and configure the refresh and hold timers:

```
configure terminal
cdp enable
cdp timer 50
cdp holdtime 100
```

CHAPTER 12

Configuring LLDP

This chapter describes how to configure the Link Layer Discovery Protocol (LLDP) in order to discover other devices on the local network.

Note For complete syntax and usage information for the commands used in this chapter, see the command reference for this release and the “System Management Commands” section in the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2*.

This chapter contains the following sections:

About LLDP

The Cisco Discovery Protocol (CDP) is a device discovery protocol that allows network management applications to automatically discover and learn about other Cisco devices that are connected to the network.

To permit the discovery of non-Cisco devices, the switch also supports the Link Layer Discovery Protocol (LLDP), a vendor-neutral device discovery protocol that is defined in the IEEE 802.1ab standard. LLDP allows network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other.

LLDP is a one-way protocol that transmits information about the capabilities and current status of a device and its interfaces. LLDP devices use the protocol to solicit information only from other LLDP devices.

LLDP supports a set of attributes that it uses to discover other devices. These attributes contain type, length, and value (TLV) descriptions. LLDP devices can use TLVs to send and receive information to other devices on the network. Details such as configuration information, device capabilities, and device identity can be advertised using this protocol.

LLDP advertises the following TLVs by default:

- Management address
- Port description
- Port VLAN
- System capabilities
- System description
- System name

High Availability

The LLDP feature supports stateless and stateful restarts. After a reboot or supervisor switchover, the running configuration is applied.

For more information on high availability, see the *Cisco Nexus Series NX-OS High Availability and Redundancy Guide*.

Virtualization Support

Only one instance of LLDP is supported in the Cisco Nexus® 3550-T switches.

Guidelines and Limitations for LLDP

LLDP has the following configuration guidelines and limitations:

- LLDP must be enabled on the device before you can enable or disable it on any interfaces.
- LLDP is supported only on physical interfaces.
- LLDP can discover up to one device per port.

Default Settings for LLDP

This table lists the LLDP default settings.

Parameters	**Default**
Global LLDP	Disabled
LLDP on interfaces	Enabled, after LLDP is enabled globally
LLDP hold time (before discarding)	120 seconds
LLDP reinitialization delay	2 seconds
LLDP timer (packet update frequency)	30 seconds
LLDP receive	Enabled, after LLDP is enabled globally
LLDP transmit	Enabled, after LLDP is enabled globally

Configuring LLDP

This chapter describes how to configure the Link Layer Discovery Protocol (LLDP) on the Cisco Nexus® 3550-T switch.

Enabling or Disabling LLDP Globally

You can enable or disable LLDP globally on a device. You must enable LLDP globally to allow a device to send and receive LLDP packets.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **feature lldp** **Example:** `switch(config)# feature lldp`	Enables or disables LLDP on the device. LLDP is disabled by default.
Step 3	(Optional) **show running-config lldp** **Example:** `switch(config)# show running-config lldp`	Displays the global LLDP configuration. If LLDP is enabled, it shows "feature lldp." If LLDP is disabled, it shows an "Invalid command" error.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling LLDP on an Interface

After you globally enable LLDP, it is enabled on all supported interfaces by default. However, you can enable or disable LLDP on individual interfaces or selectively configure an interface to only send or only receive LLDP packets.

Before you begin

Make sure that you have globally enabled LLDP on the device.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *interface slot*/*port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Specifies the interface on which you are enabling LLDP and enters the interface configuration mode.

Step 3	[**no**] **lldp transmit** **Example:** `switch(config-if)# lldp transmit`	Enables or disables the transmission of LLDP packets on an interface. After you globally enable LLDP, it is enabled on all supported interfaces by default.
Step 4	[**no**] **lldp receive** **Example:** `switch(config-if)# lldp receive`	Enables or disables the reception of LLDP packets on an interface. After you globally enable LLDP, it is enabled on all supported interfaces by default.
Step 5	(Optional) **show lldp interface** *interface slot*/*port* **Example:** `switch(config-if)# show lldp interface ethernet 1/1`	Displays the LLDP configuration on the interface.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Multiple LLDP Neighbors Per Physical Interface

Often times a network device sends multiple LLDP packets, out of which one is from the actual host. If a Cisco Nexus switch is communicating with the device but can only manage a single LLDP neighbor per interface, there is a good chance that becoming a neighbor with the actual required host will fail. To minimize this, Cisco Nexus switch interfaces can support multiple LLDP neighbors creating a better opportunity of becoming an LLDP neighbor with the correct device.

Support for multiple LLDP neighbors over the same interface requires LLDP multi-neighbor support to be configured globally.

Enabling or Disabling LLDP Multi-Neighbor Support

Before you begin

Consider the following before enabling LLDP multi-neighbor support on the interfaces:

- Make sure that you have globally enabled LLDP on the device (global configuration command **feature lldp**).

Note After you globally enable LLDP, it is enabled on all supported interfaces by default.

- A maximum of three (3) neighbors are supported on an interface.

Procedure

	Command or Action	Purpose

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

Step 2	Required: [**no**] **lldp multi-neighbor**
	Example:
	`switch(config)# lldp multi-neighbor`
	`switch(config)#`
		Enables or disables LLDP multi-neighbor support for all interfaces globally.

Step 3	**interface** *port* / *slot*
	Example:
	`switch(config)# interface 1/1`
	`switch(config-if)#`
		Specifies the interface on which you are enabling LLDP and enters the interface configuration mode.

Step 4	(Optional) [**no**] **lldp transmit** **Example:** `switch(config-if)# lldp transmit`	Disables (or enables) the transmission of LLDP packets on the interface. **Note** The transmission of LLDP packets on this interface was enabled using the global **feature lldp** command. This option is to disable the feature for this specific interface.
Step 5	(Optional) [**no**] **lldp receive** **Example:** `switch(config-if)# lldp receive`	Disables (or enables) the reception of LLDP packets on the interface. **Note** The reception of LLDP packets on this interface was enabled using the global **feature lldp** command. This option is to disable the feature for this specific interface.

	Command or Action	Purpose
Step 6	(Optional) **show lldp interface**l *port* / *slot* **Example:** `switch(config-if)# show lldp interface 1/1`	Displays the LLDP configuration on the interface.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling LLDP Support on Port-Channel Interfaces

Before you begin

Consider the following before enabling LLDP support on port-channels:

- Make sure that you have globally enabled LLDP on the device (global configuration command **feature lldp**).

Note After you globally enable LLDP, it is enabled on all supported interfaces by default.

- Applying the **lldp transmit** and **lldp receive** configuration commands to a port-channel does not affect the configuration for the members of the port-channel.
- LLDP neighbors form between the port-channels only when LLDP transmit and receive is configured on both sides of the port-channel.

Note The LLDP transmit and receive commands do not work on MCT, VPC, fex-fabric, FEX port-channels, and port-channel sub-interfaces.

If you enable the LLDP port-channel feature globally, the LLDP configuration is not applied to any of these port types. If the configuration is removed from the port-channels or the port type feature is disabled globally, you cannot use the **lldp port-channel** command to enable it on the newly supported port-channels. The command was already issued. To enable LLDP port-channel on the port-channels in question, configure **lldp transmit** and **lldp receive** for each port-channel (see steps 4, 5, and 6 in the following procedure).

Procedure

	Command or Action	Purpose

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

Step 2	Required: [**no**] **lldp port-channel**
	Example:
	`switch(config)# lldp port-channel`
	`switch(config)#`
		Enables or disables LLDP transmit and receive for all port channels globally.

Step 3	**interface port-channel** [*port-channel-number* | *port-channel-range*]
	Example:
	`switch(config)# interface port-channel 3`
	`switch(config-if)#`
	Example:
	Enter a range of port-channel numbers if you are configuring LLDP over more than one port-channel:
	`switch(config)# interface port-channel 1-3`
	`switch(config-if-range)#`
		Specifies the interface port-channel on which you are enabling LLDP and enters the interface configuration mode.
		Specifies the interface port-channel range on which you are enabling LLDP and enters the interface range configuration mode.

Step 4	(Optional) [**no**] **lldp transmit** **Example:** `switch(config-if)# lldp transmit`	Disables (or enables) the transmission of LLDP packets on the port-channel or range of port-channels. **Note** The transmission of LLDP packets on this port-channel was enabled using the global **lldp port-channel** command in step 3. This option is to disable the feature for this specific port-channel.
Step 5	(Optional) [**no**] **lldp receive** **Example:** `switch(config-if)# lldp receive`	Disables (or enables) the reception of LLDP packets on the port-channel or range of port-channels. **Note** The reception of LLDP packets on this port-channel was enabled using the global **lldp port-channel** command in step 3. This option is to disable the feature for this specific port-channel.

	Command or Action	Purpose
Step 6	(Optional) **show lldp interface port-channel**l *port-channel-number* **Example:** `switch(config-if)# show lldp interface port-channel 3`	Displays the LLDP configuration on the port-channel.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Optional LLDP Parameters

You can configure the frequency of LLDP updates, the amount of time for a receiving device to hold the information before discarding it, and the initialization delay time.

Procedure

	Command or Action	Purpose

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

Step 2	(Optional) [**no**] **lldp holdtime** *seconds* **Example:** `switch(config)# lldp holdtime 200`	Specifies the amount of time in seconds that a receiving device should hold the information that is sent by your device before discarding it. The range is 10 to 255 seconds; the default is 120 seconds.
Step 3	(Optional) [**no**] **lldp reinit** *seconds* **Example:** `switch(config)# lldp reinit 5`	Specifies the delay time in seconds for LLDP to initialize on any interface. The range is 1 to 10 seconds; the default is 2 seconds.
Step 4	(Optional) [**no**] **lldp timer** *seconds* **Example:** `switch(config)# lldp timer 50`	Specifies the transmission frequency of LLDP updates in seconds. The range is 5 to 254 seconds; the default is 30 seconds.
Step 5	(Optional) **show lldp timers** **Example:** `switch(config)# show lldp timers`	Displays the LLDP hold time, delay time, and update frequency configuration.

	Command or Action	Purpose
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the LLDP Configuration

To display the LLDP configuration, perform one of the following tasks:

Command	Purpose
show running-config lldp	Displays the global LLDP configuration.
show lldp interface *interface slot*/*port*	Displays the LLDP interface configuration.
show lldp timers	Displays the LLDP hold time, delay time, and update frequency configuration.
show lldp neighbors {**detail** | **interface** *interface slot*/*port*}	Displays the LLDP neighbor device status.
show lldp traffic interface *interface slot*/*port*	Displays the number of LLDP packets sent and received on the interface.

Use the **clear lldp counters** command to clear the LLDP statistics.

Configuration Example for LLDP

This example shows how to enable LLDP on a device; disable LLDP on some interfaces; configure optional parameters such as hold time, delay time, and update frequency:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# feature lldp
switch(config)# interface ethernet 1/9
switch(config-if)# no lldp transmit
switch(config-if)# no lldp receive
switch(config-if)# exit
switch(config)# interface ethernet 1/10
switch(config-if)# no lldp transmit
switch(config-if)# no lldp receive
switch(config-if)# exit
switch(config)# lldp holdtime 200
switch(config)# lldp reinit 5
switch(config)# lldp timer 50
```

PART IV

Cisco Nexus 3550-T Multicast Routing Configuration Guide

CHAPTER 13

Multicast Configuration Overview

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

About Multicast

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

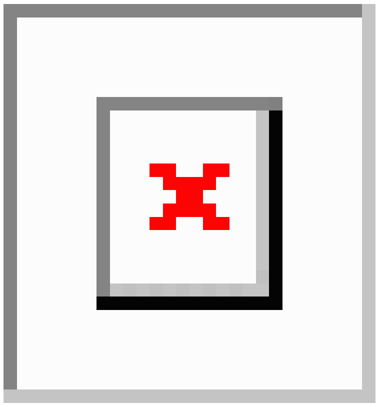
Multicast involves both a method of delivery and discovery of senders and receivers of multicast data, which is transmitted on IP multicast addresses called groups. A multicast address that includes a group and source IP address is often referred to as a channel. The Internet Assigned Number Authority (IANA) has assigned 224.0.0.0 through 239.255.255.255 as IPv4 multicast addresses. For more information, see http://www.iana.org/assignments/multicast-addresses.

Note For a complete list of RFCs related to multicast, see the *IETF RFCs for IP Multicast* chapter.

The routers in the network listen for receivers to advertise their interest in receiving multicast data from selected groups. The routers then replicate and forward the data from sources to the interested receivers. Multicast data for a group is transmitted only to those LAN segments with receivers that requested it.

This figure shows one source transmitting multicast data that is delivered to two receivers. In the figure, because the center host is on a LAN segment where no receiver requested multicast data, no data is delivered to that receiver.

Figure 4: Multicast Traffic from One Source to Two Receivers



Cisco NX-OS PIM

Cisco NX-OS supports multicasting with Protocol Independent Multicast (PIM) sparse mode. PIM is IP routing protocol independent and can leverage whichever unicast routing protocols are used to populate the unicast routing table. In PIM sparse mode, multicast traffic is sent only to locations of the network that specifically request it. PIM dense mode is not supported by Cisco NX-OS.

Note In this publication, the term “PIM” is used for PIM sparse mode version 2.

To access multicast commands, you must enable the PIM feature. Multicast is enabled only after you enable PIM on an interface of each router in a domain. You can configure PIM for an IPv4 network. By default, IGMP is running on the system.

PIM, which is used between multicast-capable routers, advertises group membership across a routing domain by constructing multicast distribution trees. PIM builds shared distribution trees, on which packets from multiple sources are forwarded, as well as source distribution trees, on which packets from a single source are forwarded.

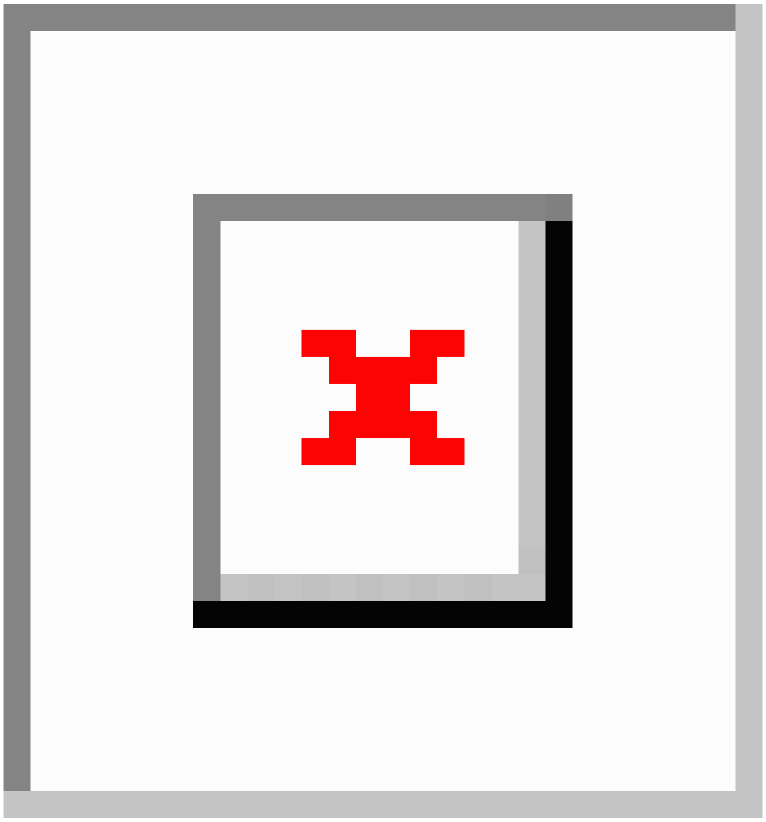
The distribution trees change automatically to reflect the topology changes due to link or router failures. PIM dynamically tracks both multicast-capable sources and receivers.

The router uses the unicast routing table and RPF routes for multicast to create multicast routing information.

Note In this publication, “PIM for IPv4” refers to the Cisco NX-OS implementation of PIM sparse mode.

This figure shows two PIM domains in an IPv4 network.

Figure 5: PIM Domains in an IPv4 Network



- The lines with arrows show the path of the multicast data through the network. The multicast data originates from the sources at hosts A and D.
- The dashed line connects routers B and F, which are Multicast Source Discovery Protocol (MSDP) peers. MSDP supports the discovery of multicast sources in other PIM domains.
- Hosts B and C receive multicast data by using Internet Group Management Protocol (IGMP) to advertise requests to join a multicast group.
- Routers A, C, and D are designated routers (DRs). When more than one router is connected to a LAN segment, such as C and E, the PIM software chooses one router to be the DR so that only one router is responsible for putting multicast data on the segment.

Router B is the rendezvous point (RP) for one PIM domain, and router F is the RP for the other PIM domain. The RP provides a common point for connecting sources and receivers within a PIM domain.

PIM only supports Any source multicast (ASM) mode for connecting sources and receivers.

ASM

Any Source Multicast (ASM) is a PIM tree building mode that uses shared trees to discover new sources and receivers as well as source trees to form shortest paths from receivers to sources. The shared tree uses a network node as the root, called the rendezvous point (RP). The source tree is rooted at first-hop routers, directly attached to each source that is an active sender. The ASM mode requires an RP for a group range. An RP can be configured statically. If an RP is learned, the group operates in ASM mode.

The ASM mode is the default mode when you configure RPs.

IGMP

By default, the Internet Group Management Protocol (IGMP) for PIM is running on the system.

IGMP is used by hosts that want to receive multicast data to request membership in multicast groups. Once the group membership is established, multicast data for the group is directed to the LAN segment of the requesting host.

You can configure IGMPv2 on an interface. By default, the software enables IGMPv2.

Note There are limitations to using IGMPv2 on Layer 2 ports with PIM disabled. Please see the Guidelines and Limitations for IGMP Snooping, on page 169 section before using the feature.

Guidelines and Limitations for Multicast

- Cisco Nexus® 3550-T platform cannot support FHR.
- Multicast over Trunk is not supported in Cisco Nexus® 3550-T platform.
- Traffic storm control is not supported for unknown multicast traffic.
- Bidirectional mode is not supported on Cisco Nexus® 3550-T platform switches.
- Cisco Nexus® 3550-T does not do the multicast RPF check. RPF failed packets are flooded to learned receivers.

High-Availability Requirements for Multicast

After a multicast routing protocol is restarted, its state is recovered from the MRIB process. When a supervisor switchover occurs, the MRIB recovers its state from the hardware, and the multicast protocols recover their state from periodic message activity.

Troubleshooting Inconsistency Between SW and HW Multicast Routes

Symptom

This section provides symptoms, possible causes, and recommended actions for when *, G, entries that are seen in the MRIB with active flow, but are not programmed in MFIB.

Possible Cause

The issue can be seen when numerous active flows are received beyond the hardware capacity. This causes some of the entries not to be programmed in hardware while there is no free hardware index.

If the number of active flows are significantly reduced to free up the hardware resource, inconsistency may be seen between MRIB and MFIB for flows that were previously affected when the hardware table was full until the entry, times out, repopulates, and triggers programming.

There is currently no mechanism to walk the MRIB table and reprogram missing entries in HW after hardware resource is freed.

Corrective Action

To ensure reprogramming of the entries, use the **clear ip mroute *** command.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

CHAPTER 14

Configuring IGMP

This chapter describes how to configure the Internet Group Management Protocol (IGMP) on Cisco NX-OS devices for IPv4 networks.

About IGMP

IGMP is an IPv4 protocol that a host uses to request multicast data for a particular group. Using the information obtained through IGMP, the software maintains a list of multicast group or channel memberships on a per-interface basis. The systems that receive these IGMP packets send multicast data that they receive for requested groups or channels out the network segment of the known receivers.

By default, the IGMP process is running. You cannot enable IGMP manually on an interface. IGMP is automatically enabled when you perform one of the following configuration tasks on an interface:

- Enable PIM
- Statically bind a local multicast group

IGMP Versions

The device supports IGMPv2 and IGMPv3, and IGMPv1 report reception.

By default, the software enables IGMPv2 when it starts the IGMP process. You can enable IGMPv3 on interfaces where you want its capabilities.

IGMPv3 includes the following key changes from IGMPv2:

- Hosts no longer perform report suppression, which means that hosts always send IGMP membership reports when an IGMP query message is received.

Note The Cisco Nexus® 3550-T switches does not support SSM.

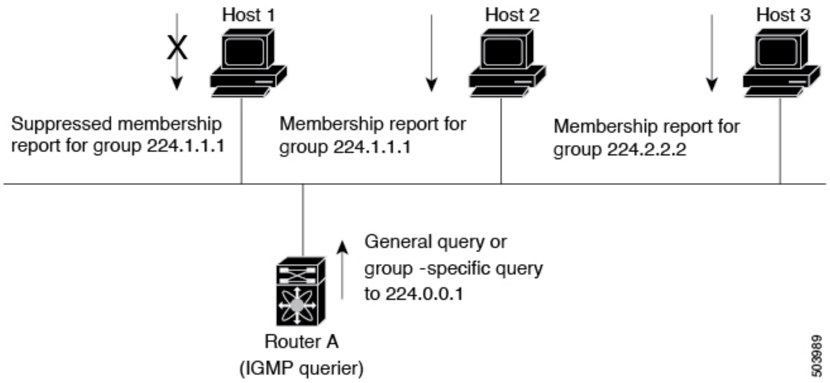
For detailed information about IGMPv2, see RFC 2236.

For detailed information about IGMPv3, see RFC 5790.

IGMP Basics

This figure shows the basic IGMP process of a router that discovers multicast hosts. Hosts 1, 2, and 3 send unsolicited IGMP membership report messages to initiate receiving multicast data for a group or channel.

Figure 6: IGMPv1 and IGMPv2 Query-Response Process


Host 1
Host 2
Host 3
X
Suppressed membership report for group 224.1.1.1
Membership report for group 224.1.1.1
Membership report for group 224.2.2.2
General query or group -specific query to 224.0.0.1
Router A (IGMP querier)
503989

In the figure below, router A, which is the IGMP designated querier on the subnet, sends query messages to the all-hosts multicast group at 224.0.0.1 periodically to discover whether any hosts want to receive multicast data. You can configure the group membership timeout value that the router uses to determine that no members of a group or source exist on the subnet.

The software elects a router as the IGMP querier on a subnet if it has the lowest IP address. As long as a router continues to receive query messages from a router with a lower IP address, it resets a timer that is based on its querier timeout value. If the querier timer of a router expires, it becomes the designated querier. If that router later receives a host query message from a router with a lower IP address, it drops its role as the designated querier and sets its querier timer again.

In this figure, host 1's membership report is suppressed, and host 2 sends its membership report for group 224.1.1.1 first. Host 1 receives the report from host 2. Because only one membership report per group needs to be sent to the router, other hosts suppress their reports to reduce network traffic. Each host waits for a random time interval to avoid sending reports at the same time. You can configure the query maximum response time parameter to control the interval in which hosts randomize their responses.

Note IGMPv1 and IGMPv2 membership report suppression occurs only on hosts that are connected to the same port.

Note IGMPv3 hosts do not perform IGMP membership report suppression.

Messages sent by the designated querier have a time-to-live (TTL) value of 1, which means that the messages are not forwarded by the directly connected routers on the subnet. You can configure the frequency and number of query messages sent specifically for IGMP startup, and you can configure a short query interval at startup so that the group state is established as quickly as possible. Although usually unnecessary, you can tune the query interval used after startup to a value that balances the responsiveness to host group membership messages and the traffic created on the network.

Caution Changing the query interval can severely impact multicast forwarding.

When a multicast host leaves a group, a host that runs IGMPv2 or later sends an IGMP leave message. To check if this host is the last host to leave the group, the software sends an IGMP query message and starts a timer that you can configure called the last member query response interval. If no reports are received before the timer expires, the software removes the group state. The router continues to send multicast traffic for a group until its state is removed.

You can configure a robustness value to compensate for packet loss on a congested network. The robustness value is used by the IGMP software to determine the number of times to send messages.

Link local addresses in the range 224.0.0.0/24 are reserved by the Internet Assigned Numbers Authority (IANA). Network protocols on a local network segment use these addresses; routers do not forward these addresses because they have a TTL of 1. By default, the IGMP process sends membership reports only for nonlink local addresses, but you can configure the software to send reports for link local addresses.

Prerequisites for IGMP

IGMP has the following prerequisites:

- You are logged onto the device.
- For global configuration commands. The default configuration mode shown in the examples in this chapter applies to the default VRF.

Guidelines and Limitations for IGMP

IGMP has the following guidelines and limitations:

- For having low latency, Cisco Nexus® 3550-T switch only supports {Vlan,MAC} lookup for L2 ports. There is no IP based {VLAN,G} or {VLAN,G,S} lookup.
- Route-Aliasing is expected as routes are installed for optimized {Vlan,MAC} lookup.
- All unknown multicast packet miss is dropped instead of flood to other routers in the in the same L2 Domain.

- Multi-access Network with Cisco Nexus® 3550-T switch would not work, there cannot be 2 PIM-Routers in same VLAN segment if one of the PIM enabled routers is Cisco Nexus® 3550-T switch. Cisco Nexus® 3550-T switch cannot act as non-DR.
- PIM cannot be enabled on L2 Transit node since multicast-lookup miss packets not flooded in VLAN.
- Cisco Nexus® 3550-T switch should not be used as L3 transit box on SVIs. Though L2 receivers may be present on the transit box on SVI. It can be used as L3 transit on L3 physical ports.
- No FHR Support – hence no source expected to be connected directly in VLAN where L3 Multicast routing is required.
- Owing to {Vlan,Mac} lookup, IGMPv2 reports are flooded to the receivers already attached, this results in report-suppression. It is recommended to have hosts configured as IGMPv3.
- Excluding or blocking a list of sources according to IGMPv3 (RFC 5790) is not supported.

Default Settings for IGMP

This table lists the default settings for IGMP parameters.

Table 11: Default IGMP Parameters

Parameters	Default
IGMP version	2
Startup query interval	30 seconds
Startup query count	2
Robustness value	2
Querier timeout	255 seconds
Query timeout	255 seconds
Query max response time	10 seconds
Query interval	125 seconds
Last member query response interval	1 second
Last member query count	2
Group membership timeout	260 seconds
Report link local multicast groups	Disabled
Enforce router alert	Disabled
Immediate leave	Disabled

Configuring IGMP Parameters

You can configure the IGMP global and interface parameters to affect the operation of the IGMP process.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Configuring IGMP Interface Parameters

You can configure the optional IGMP interface parameters described in the table below.

Table 12: IGMP Interface Parameters

Parameter	Description
IGMP version	IGMP version that is enabled on the interface. The IGMP version can be 2 or 3. The default is 2.
Static multicast groups	Multicast groups that are statically bound to the interface. You can configure the groups to join the interface with the (*, G) state. You can specify a route-map policy name that lists the group prefixes, and group ranges to use with the **match ip multicast** command. **Note** Although you can configure the (*, G) state, the source tree is built only if you enable IGMPv3. You can configure a multicast group on all the multicast-capable routers on the network so that pinging the group causes all the routers to respond.
Static multicast groups on OIF	Multicast groups that are statically bound to the output interface. You can configure the groups to join the output interface with the (*, G) state or specify a source IP to join with the (*, G) state. You can specify a route-map policy name that lists the group prefixes, group ranges, and source prefixes to use with the **match ip multicast** command. **Note** Although you can configure the (*, G) state, the source tree is built only if you enable IGMPv3.

Parameter	Description
Startup query interval	Startup query interval. By default, this interval is shorter than the query interval so that the software can establish the group state as quickly as possible. Values range from 1 to 18,000 seconds. The default is 31 seconds.
Startup query count	Number of queries sent at startup that are separated by the startup query interval. Values range from 1 to 10. The default is 2.
Robustness value	Robustness variable that you can tune to reflect expected packet loss on a congested network. You can increase the robustness variable to increase the number of times that packets are resent. Values range from 1 to 7. The default is 2.
Querier timeout	Number of seconds that the software waits after the previous querier has stopped querying and before it takes over as the querier. Values range from 1 to 65,535 seconds. The default is 255 seconds.
Query max response time	Maximum response time advertised in IGMP queries. You can tune the IGMP messages on the network by setting a larger value so that host responses are spread out over a longer time. This value must be less than the query interval. Values range from 1 to 25 seconds. The default is 10 seconds.
Query interval	Frequency at which the software sends IGMP host query messages. You can tune the number of IGMP messages on the network by setting a larger value so that the software sends IGMP queries less often. Values range from 1 to 18,000 seconds. The default is 125 seconds.
Last member query response interval	Interval in which the software sends a response to an IGMP query after receiving a host leave message from the last known active host on the subnet. If no reports are received in the interval, the group state is deleted. You can use this value to tune how quickly the software stops transmitting on the subnet. The software can detect the loss of the last member of a group or source more quickly when the values are smaller. Values range from 1 to 25 seconds. The default is 1 second.

Parameter	Description
Last member query count	Number of times that the software sends an IGMP query, separated by the last member query response interval, in response to a host leave message from the last known active host on the subnet. Values range from 1 to 5. The default is 2. Setting this value to 1 means that a missed packet in either direction causes the software to remove the multicast state from the queried group or channel. The software may wait until the next query interval before the group is added again.
Group membership timeout	Group membership interval that must pass before the router decides that no members of a group or source exist on the network. Values range from 3 to 65,535 seconds. The default is 260 seconds.
Report link local multicast groups	Option that enables sending reports for groups in 224.0.0.0/24. Link local addresses are used only by protocols on the local network. Reports are always sent for nonlink local groups. The default is disabled.
Immediate leave	Option that minimizes the leave latency of IGMPv2 group memberships on a given IGMP interface because the device does not send group-specific queries. When immediate leave is enabled, the device removes the group entry from the multicast routing table immediately upon receiving a leave message for the group. The default is disabled. **Note** Use this command only when there is one receiver behind the interface for a given group.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *interface*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters interface configuration mode.
		Note Use the commands listed from step-3 to configure the IGMP interface parameters.

	Command or Action	Purpose
Step 3	**ip igmp version** *value* **Example:** `switch(config-if)# ip igmp version 3`	Sets the IGMP version to the value specified. Values can be 2 or 3. The default is 2. The **no** form of the command sets the version to 2.

Step 4	**ip igmp join-group** {*group* [**source** *source*] }
	Example:
	`switch(config-if)# ip igmp join-group 230.0.0.0`
		Configures an interface on the device to join the specified group or channel. The device accepts the multicast packets for CPU consumption only.
		Caution The device CPU must be able to handle the traffic generated by using this command. Because of CPU load constraints, using this command, especially in any form of scale, is not recommended. Consider using the **ip igmp static-oif** command instead.

Step 5	**ip igmp static-oif** {*group* [**source** *source*]}
	Example:
	`switch(config-if)# ip igmp static-oif 230.0.0.0`
		Statically binds a multicast group to the outgoing interface, which is handled by the device hardware. If you specify only the group address, the (*, G) state is created. If you specify the source address, the (*, G) state is created.
		Note A source tree is built for the (*, G) state only if you enable IGMPv3.

Step 6	**ip igmp startup-query-interval** *seconds* **Example:** `switch(config-if)# ip igmp startup-query-interval 25`	Sets the query interval used when the software starts up. Values can range from 1 to 18,000 seconds. The default is 31 seconds.
Step 7	**ip igmp startup-query-count** *count* **Example:** `switch(config-if)# ip igmp startup-query-count 3`	Sets the query count used when the software starts up. Values can range from 1 to 10. The default is 2.
Step 8	**ip igmp robustness-variable** *value* **Example:** `switch(config-if)# ip igmp robustness-variable 3`	Sets the robustness variable. Values can range from 1 to 7. The default is 2.
Step 9	**ip igmp querier-timeout** *seconds* **Example:** `switch(config-if)# ip igmp querier-timeout 300`	Sets the querier timeout that the software uses when deciding to take over as the querier. Values can range from 1 to 65,535 seconds. The default is 255 seconds.

	Command or Action	Purpose
Step 10	**ip igmp query-timeout** *seconds* **Example:** `switch(config-if)# ip igmp query-timeout 300`	Sets the query timeout that the software uses when deciding to take over as the querier. Values can range from 1 to 65,535 seconds. The default is 255 seconds. **Note** This command has the same functionality as the **ip igmp querier-timeout** command.
Step 11	**ip igmp query-max-response-time** *seconds* **Example:** `switch(config-if)# ip igmp query-max-response-time 15`	Sets the response time advertised in IGMP queries. Values can range from 1 to 25 seconds. The default is 10 seconds.
Step 12	**ip igmp query-interval** *interval* **Example:** `switch(config-if)# ip igmp query-interval 100`	Sets the frequency at which the software sends IGMP host query messages. Values can range from 1 to 18,000 seconds. The default is 125 seconds.
Step 13	**ip igmp last-member-query-response-time** *seconds* **Example:** `switch(config-if)# ip igmp last-member-query-response-time 3`	Sets the query interval waited after sending membership reports before the software deletes the group state. Values can range from 1 to 25 seconds. The default is 1 second.
Step 14	**ip igmp last-member-query-count** *count* **Example:** `switch(config-if)# ip igmp last-member-query-count 3`	Sets the number of times that the software sends an IGMP query in response to a host leave message. Values can range from 1 to 5. The default is 2.
Step 15	**ip igmp group-timeout** *seconds* **Example:** `switch(config-if)# ip igmp group-timeout 300`	Sets the group membership timeout for IGMPv2. Values can range from 3 to 65,535 seconds. The default is 260 seconds.
Step 16	**ip igmp report-link-local-groups** **Example:** `switch(config-if)# ip igmp report-link-local-groups`	Enables sending reports for groups in 224.0.0.0/24. Reports are always sent for nonlink local groups. By default, reports are not sent for link local groups.
Step 17	**ip igmp report-policy** *policy* **Example:** `switch(config-if)# ip igmp report-policy my_report_policy`	Configures an access policy for IGMP reports that is based on a route-map policy.

	Command or Action	Purpose
Step 18	**ip igmp access-group** *policy* **Example:** `switch(config-if)# ip igmp access-group my_access_policy`	Configures a route-map policy to control the multicast groups that hosts on the subnet serviced by an interface can join. **Note** Only the **match ip multicast group** command is supported in this route map policy. The **match ip address** command for matching an ACL is not supported.
Step 19	**ip igmp immediate-leave** **Example:** `switch(config-if)# ip igmp immediate-leave`	Enables the device to remove the group entry from the multicast routing table immediately upon receiving a leave message for the group. Use this command to minimize the leave latency of IGMPv2 group memberships on a given IGMP interface because the device does not send group-specific queries. The default is disabled. **Note** Use this command only when there is one receiver behind the interface for a given group.
Step 20	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Restarting the IGMP Process

You can restart the IGMP process and optionally flush all routes.

Procedure

	Command or Action	Purpose
Step 1	**restart igmp** **Example:** `switch# restart igmp`	Restarts the IGMP process.

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

	Command or Action	Purpose
Step 3	**ip igmp flush-routes** **Example:** `switch(config)# ip igmp flush-routes`	Removes routes when the IGMP process is restarted. By default, routes are not flushed.
Step 4	(Optional) **show running-configuration igmp** **Example:** `switch(config)# show running-configuration igmp`	Shows the running-configuration information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the IGMP Configuration

To display the IGMP configuration information, perform one of the following tasks:

Command	Description
show ip igmp interface [*interface*] [**brief**]	Displays IGMP information about all interfaces or a selected interface.
show ip igmp groups [{**source** [*group*]}] | {**group** [*source*]}] [**interface**] [**summary**]	Displays the IGMP attached group membership for a group or interface.
show ip igmp route [{**source** [*group*]}] | {**group** [*source*]}] [**interface**] [**summary**]	Displays the IGMP attached group membership for a group or interface.
show ip igmp local-groups	Displays the IGMP local group membership.
show running-configuration igmp	Displays the IGMP running-configuration information.
show startup-configuration igmp	Displays the IGMP startup-configuration information.

Configuration Examples for IGMP

The following example shows how to configure the IGMP parameters:

```
configure terminal

  interface ethernet 1/1
    ip igmp version 3
    ip igmp join-group 230.0.0.0
    ip igmp startup-query-interval 25
    ip igmp startup-query-count 3
    ip igmp robustness-variable 3
    ip igmp querier-timeout 300
    ip igmp query-timeout 300
    ip igmp query-max-response-time 15
    ip igmp query-interval 100
    ip igmp last-member-query-response-time 3
    ip igmp last-member-query-count 3
    ip igmp group-timeout 300
    ip igmp report-link-local-groups
```

CHAPTER 15

Configuring IGMP Snooping

This chapter describes how to configure the Internet Group Management Protocol (IGMP) Snooping on Cisco NX-OS devices for IPv4 networks.

About IGMP Snooping

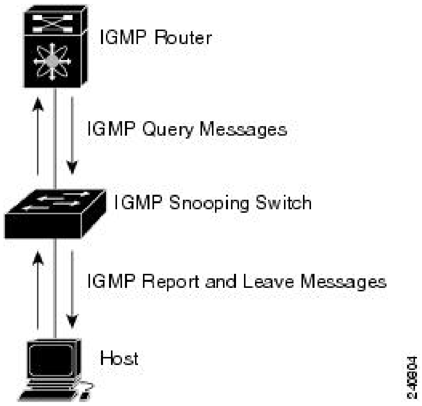
Note We recommend that you do not disable IGMP snooping on the device. If you disable IGMP snooping, you might see reduced multicast performance because of excessive false flooding within the device.

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

This figure shows an IGMP snooping switch that sits between the host and the IGMP router. The IGMP snooping switch snoops the IGMP membership reports and Leave messages and forwards them only when necessary to the connected IGMP routers.

Note Owing to {Vlan,Mac} lookup IGMPv2 reports are flooded/forwarded to the receivers already attached, this results in report-suppression. This is specific to Cisco Nexus 3550-T 10.1(2t) release only.

Figure 7: IGMP Snooping Switch



The IGMP snooping software operates upon IGMPv1, IGMPv2, and IGMPv3 control plane packets where Layer 3 control plane packets are intercepted and influence the Layer 2 forwarding behavior.

The Cisco NX-OS IGMP snooping software has the following proprietary features:

- Multicast forwarding based on the MAC address

For more information about IGMP snooping, see RFC4541

IGMPv1 and IGMPv2

Both IGMPv1 and IGMPv2 support membership report suppression, which means that if two hosts on the same subnet want to receive multicast data for the same group, the host that receives a member report from the other host suppresses sending its report. Membership report suppression occurs for hosts that share a port.

If no more than one host is attached to each VLAN switch port, you can configure the fast leave feature in IGMPv2. The fast leave feature does not send last member query messages to hosts. As soon as the software receives an IGMP leave message, the software stops forwarding multicast data to that port.

IGMPv1 does not provide an explicit IGMP leave message, so the software must rely on the membership message timeout to indicate that no hosts remain that want to receive multicast data for a particular group.

Note The software ignores the configuration of the last member query interval when you enable the fast leave feature because it does not check for remaining hosts.

IGMP Snooping Querier

When PIM is not enabled on an interface because the multicast traffic does not need to be routed, you must configure an IGMP snooping querier to send membership queries. You define the querier in a VLAN that contains multicast sources and receivers but no other active querier.

The querier can be configured to use any IP address in the VLAN.

As a best practice, a unique IP address, one that is not already used by the switch interface or the Hot Standby Router Protocol (HSRP) virtual IP address, should be configured so as to easily reference the querier.

Note The IP address for the querier should not be a broadcast IP address, multicast IP address, or 0 (0.0.0.0).

When an IGMP snooping querier is enabled, it sends out periodic IGMP queries that trigger IGMP report messages from hosts that want to receive IP multicast traffic. IGMP snooping listens to these IGMP reports to establish appropriate forwarding.

The IGMP snooping querier performs querier election as described in RFC 2236. Querier election occurs in the following configurations:

- When there are multiple switch queriers configured with the same subnet on the same VLAN on different switches.
- When the configured switch querier is in the same subnet as with other Layer 3 SVI queriers.

Prerequisites for IGMP Snooping

IGMP snooping has the following prerequisites:

- You are logged onto the device.
- The default configuration mode shown in the examples in this chapter applies to the default VRF.

Guidelines and Limitations for IGMP Snooping

IGMP snooping has the following guidelines and limitations:

- Cisco Nexus® 3550-T switch forwards known multicast packets based on Multicast DestMAC of packets on incoming ports where PIM is disabled to provide lower latency. Hence, Cisco Nexus® 3550-T switch IGMPv1/v2 incoming reports are forwarded to known multicast receivers.
- For having low latency, Cisco Nexus® 3550-T switch only supports {Vlan,MAC} lookup for L2 ports. There is no IP based {VLAN,G} or {VLAN,G,S} lookup.
- Owing to {Vlan,Mac} lookup, IGMPv2 reports are flooded to the receivers already attached, this results in report-suppression. It is recommended to have hosts configured as IGMPv3.
- You must enable the **ip igmp snooping group-timeout** command when you use the **ip igmp snooping proxy general-queries** command. We recommend that you set it to "never". Otherwise, you might experience multicast packet loss.
- Cisco Nexus® 3550-T switches support IGMP snooping for IPv4 but do not support MLD snooping for IPv6.
- Layer 3 IPv6 multicast routing is not supported.

Default Settings

Parameters	Default
IGMP snooping	Enabled
Explicit tracking	Enabled
Fast leave	Disabled
Last member query interval	1 second
Snooping querier	Disabled
Report suppression	Enabled
Link-local groups suppression	Enabled
Optimise-multicast-flood	Disabled
IGMPv3 report suppression for the entire device	Disabled
IGMPv3 report suppression per VLAN	Enabled

Configuring IGMP Snooping Parameters

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Note You must enable IGMP snooping globally before any other commands take effect.

Configuring Global IGMP Snooping Parameters

To affect the operation of the IGMP snooping process globally, you can configure various optional IGMP snooping parameters.

Notes for IGMP Snooping Parameters

- IGMP Snooping Proxy parameter

  To decrease the burden placed on the snooping switch during each IGMP general query (GQ) interval, the Cisco NX-OS software provides a way to decouple the periodic general query behavior of the IGMP snooping switch from the query interval configured on the multicast routers.

  You can configure the device to consume IGMP general queries from the multicast router, rather than flooding the general queries to all the switchports. When the device receives a general query, it produces proxy reports for all currently active groups and distributes the proxy reports over the period specified by the MRT that is specified in the router query. At the same time, independent of the periodic general query activity of the multicast router, the device sends an IGMP general query on each port in the VLAN in a round-robin fashion. It cycles through all the interfaces in the VLAN at the rate given by the following formula.

Rate = {number of interfaces in VLAN} * {configured MRT} * {number of VLANs}

When queries are run in this mode, the default MRT value is 5,000 milliseconds (5 seconds). For a device that has 500 switchports in a VLAN, it would take 2,500 seconds (40 minutes) to cycle through all the interfaces in the system. This is also true when the device itself is the querier.

This behavior ensures that only one host responds to a general query at a given time, and it keeps the simultaneous reporting rate below the packet-per-second IGMP capability of the device (approximately 3,000 to 4,000 pps).

Note When you use this option, you must change the **ip igmp snooping group-timeout** parameter to a high value or to never time out.

The **ip igmp snooping proxy general-queries** [**mrt**] command causes the snooping function to proxy reply to general queries from the multicast router while also sending round-robin general queries on each switchport with the specified MRT value. (The default MRT value is 5 seconds.)

- IGMP Snooping Group-timeout parameter

  Configuring the group-timeout parameter disables the behavior of an expiring membership based on three missed general queries. Group membership remains on a given switchport until the device receives an explicit IGMP leave on that port.

  The **ip igmp snooping group-timeout** {*timeout* | **never**} command modifies or disables the behavior of an expiring IGMP snooping group membership after three missed general queries.

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Use the following commands to configure global IGMP snooping parameters.

Option	Description

`ip igmp snooping`
`switch(config)# ip igmp snooping`
	Enables IGMP snooping for the device. The default is enabled.

Option	Description
	Note If the global setting is disabled with the **no** form of this command, IGMP snooping on all VLANs is disabled, whether IGMP snooping is enabled on a VLAN or not. If you disable IGMP snooping, Cisco Nexus® 3550-Tswitch only disables IGMP snoop packet handling. Hence, even with the **no** form of this command, IGMP packets including multicast packets are not forwarded in hardware.
ip igmp snooping event-history `switch(config)# ip igmp snooping event-history`	Configures the size of the event history buffer. The default is small.
ip igmp snooping group-timeout {*minutes* | **never**} `switch(config)# ip igmp snooping group-timeout never`	Configures the group membership timeout value for all VLANs on the device.
ip igmp snooping link-local-groups-suppression `switch(config)# ip igmp snooping link-local-groups-suppression`	Configures link-local groups suppression for the entire device. The default is enabled.
ip igmp snooping proxy general-inquiries [**mrt** *seconds*] `switch(config)# ip igmp snooping proxy general-inquiries`	Configures the IGMP snooping proxy for the device. The default is 5 seconds.
ip igmp snooping v3-report-suppression `switch(config)# ip igmp snooping v3-report-suppression`	Limits the membership report traffic sent to multicast-capable routers. When you disable report suppression, all IGMP reports are sent as-is to multicast-capable routers. The default is enabled.
ip igmp snooping report-suppression `switch(config)# ip igmp snooping report-suppression`	Configures IGMPv3 report suppression and proxy reporting. The default is disabled.

Step 3 **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

(Optional) Copies the running configuration to the startup configuration.

Configuring IGMP Snooping Parameters per VLAN

To affect the operation of the IGMP snooping process per VLAN, you can configure various optional IGMP snooping parameters.

Note You configure the IGMP snooping parameters that you want by using this configuration mode; however, the configurations apply only after you specifically create the specified VLAN. See the *Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide* for information on creating VLANs.

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 **ip igmp snooping**

Example:

```
switch(config)# ip igmp snooping
```

Enables IGMP snooping. The default is enabled.

Note If the global setting is disabled with the **no** form of this command, IGMP snooping on all VLANs is disabled, whether IGMP snooping is enabled on a VLAN or not. If you disable IGMP snooping, Layer 2 multicast frames flood to all modules.

Step 3 **vlan configuration** *vlan-id*

Example:

```
switch(config)# vlan configuration 2
switch(config-vlan-config)#
```

Configures the IGMP snooping parameters you want for the VLAN. These configurations do not apply until you create the specified VLAN.

Step 4 Use the following commands to configure IGMP snooping parameters per VLAN.

Option	Description
ip igmp snooping `switch(config-vlan-config)# ip igmp snooping`	Enables IGMP snooping for the current VLAN. The default is enabled.
ip igmp snooping access-group {**prefix-list** | **route-map**} *policy-name* **interface** *interface slot/port* `switch(config-vlan-config)# ip igmp snooping access-group prefix-list plist interface ethernet 1/2`	Configures a filter for IGMP snooping reports that is based on a prefix-list or route-map policy. The default is disabled.
ip igmp snooping explicit-tracking `switch(config-vlan-config)# ip igmp snooping explicit-tracking`	Tracks IGMPv3 membership reports from individual hosts for each port on a per-VLAN basis. The default is enabled on all VLANs.
ip igmp snooping fast-leave `switch(config-vlan-config)# ip igmp snooping fast-leave`	Supports IGMPv2 hosts that cannot be explicitly tracked because of the host report suppression mechanism of the IGMPv2 protocol. When you enable fast leave, the IGMP software assumes that no more than one host is present on each VLAN port. The default is disabled for all VLANs.
ip igmp snooping group-timeout {*minutes* | **never**} `switch(config-vlan-config)# ip igmp snooping group-timeout never`	Configures the group membership timeout for the specified VLANs.
ip igmp snooping last-member-query-interval *seconds* `switch(config-vlan-config)# ip igmp snooping last-member-query-interval 3`	Removes the group from the associated VLAN port if no hosts respond to an IGMP query message before the last member query interval expires. Values range from 1 to 25 seconds. The default is 1 second.
ip igmp snooping proxy general-queries [**mrt** *seconds*] `switch(config-vlan-config)# ip igmp snooping proxy general-queries`	Configures an IGMP snooping proxy for specified VLANs. The default is 5 seconds.
ip igmp snooping querier *ip-address*	Configures a snooping querier when you do not enable PIM because multicast traffic does not need to be routed. The IP address is used as the source in messages.

Option	Description
`switch(config-vlan-config)# ip igmp snooping querier 172.20.52.106`	

`ip igmp snooping querier-timeout` *`seconds`*
`switch(config-vlan-config)# ip igmp snooping querier-timeout 300`
	Configures a snooping querier timeout value for IGMPv2 when you do not enable PIM because multicast traffic does not need to be routed. The default is 255 seconds.

`ip igmp snooping query-interval` *`seconds`*
`switch(config-vlan-config)# ip igmp snooping query-interval 120`
	Configures a snooping query interval when you do not enable PIM because multicast traffic does not need to be routed. The default value is 125 seconds.

`ip igmp snooping query-max-response-time` *`seconds`*
`switch(config-vlan-config)# ip igmp snooping query-max-response-time 12`
	Configures a snooping MRT for query messages when you do not enable PIM because multicast traffic does not need to be routed. The default value is 10 seconds.

`ip igmp snooping report-policy` `{`**`prefix-list`** `|` **`route-map`**`}` *`policy-name`* **`interface`** *`interface slot/port`*
`switch(config-vlan-config)# ip igmp snooping report-policy route-map rmap interface ethernet 1/4`
	Configures a filter for IGMP snooping reports that is based on a prefix-list or route-map policy. The default is disabled.

`ip igmp snooping startup-query-count` *`value`*
`switch(config-vlan-config)# ip igmp snooping startup-query-count 5`
	Configures snooping for a number of queries sent at startup when you do not enable PIM because multicast traffic does not need to be routed.

`ip igmp snooping startup-query-interval` *`seconds`*
`switch(config-vlan-config)# ip igmp snooping startup-query-interval 15000`
	Configures a snooping query interval at startup when you do not enable PIM because multicast traffic does not need to be routed.

`ip igmp snooping robustness-variable` *`value`*
`switch(config-vlan-config)# ip igmp snooping robustness-variable 5`
	Configures the robustness value for the specified VLANs. The default value is 2.

Option	Description
ip igmp snooping report-suppression `switch(config-vlan-config)# ip igmp snooping report-suppression`	Limits the membership report traffic sent to multicast-capable routers. When you disable report suppression, all IGMP reports are sent as-is to multicast-capable routers. The default is enabled.
ip igmp snooping mrouter interface *interface* `switch(config-vlan-config)# ip igmp snooping mrouter interface ethernet 1/1`	Configures a static connection to a multicast router. The interface to the router must be in the selected VLAN. You can specify the interface by the type and the number, such as **ethernet** *slot*/*port*.
ip igmp snooping static-group *group-ip-addr* [**source** *source-ip-addr*] **interface** *interface* `switch(config-vlan-config)# ip igmp snooping static-group 230.0.0.1 interface ethernet 1/1`	Configures the Layer 2 port of a VLAN as a static member of a multicast group. You can specify the interface by the type and the number, such as **ethernet** *slot*/*port*.
ip igmp snooping link-local-groups-suppression `switch(config-vlan-config)# ip igmp snooping link-local-groups-suppression`	Configures link-local groups suppression for the specified VLANs. The default is enabled.
ip igmp snooping v3-report-suppression `switch(config-vlan-config)# ip igmp snooping v3-report-suppression`	Configures IGMPv3 report suppression and proxy reporting for the specified VLANs. The default is enabled per VLAN.
ip igmp snooping version *value* `switch(config-vlan-config)# ip igmp snooping version 2`	Configures the IGMP version number for the specified VLANs.

Step 5 **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

(Optional) Copies the running configuration to the startup configuration.

Verifying the IGMP Snooping Configuration

Command	Description
show ip igmp snooping [**vlan** *vlan-id*]	Displays the IGMP snooping configuration by VLAN.
show ip igmp snooping groups [*source* [*group*] | *group* [*source*]] [**vlan** *vlan-id*] [**detail**]	Displays IGMP snooping information about groups by VLAN.
show ip igmp snooping querier [**vlan** *vlan-id*]	Displays IGMP snooping queriers by VLAN.
show ip igmp snooping mroute [**vlan** *vlan-id*]	Displays multicast router ports by VLAN.
show ip igmp snooping explicit-tracking [**vlan** *vlan-id*] [**detail**]	Displays IGMP snooping explicit tracking information by VLAN.

Displaying IGMP Snooping Statistics

You can display the IGMP snooping statistics using these commands.

Command	Description
show ip igmp snooping statistics vlan	Displays IGMP snooping statistics. You can see the virtual port channel (vPC) statistics in this output.
show ip igmp snooping {**report-policy** | **access-group**} **statistics** [**vlan** *vlan*]	Displays detailed statistics per VLAN when IGMP snooping filters are configured.

Clearing IGMP Snooping Statistics

You can clear the IGMP snooping statistics using these commands.

Command	Description
clear ip igmp snooping statistics vlan	Clears the IGMP snooping statistics.
clear ip igmp snooping {**report-policy** | **access-group**} **statistics** [**vlan** *vlan*]	Clears the IGMP snooping filter statistics.

Configuration Examples for IGMP Snooping

Note The configurations in this section apply only after you create the specified VLAN. See the *Cisco Nexus 3550-T Layer 2 Switching Configuration* section for information on creating VLANs.

The following example shows how to configure the IGMP snooping parameters:

```
config t
  ip igmp snooping
  vlan configuration 2
    ip igmp snooping
    ip igmp snooping explicit-tracking
    ip igmp snooping fast-leave
    ip igmp snooping last-member-query-interval 3
    ip igmp snooping querier 172.20.52.106
    ip igmp snooping report-suppression
    ip igmp snooping mrouter interface ethernet 1/1
    ip igmp snooping static-group 230.0.0.1 interface ethernet 1/1
    ip igmp snooping link-local-groups-suppression
    ip igmp snooping v3-report-suppression
```

The following example shows how to configure prefix lists and use them to filter IGMP snooping reports:

```
ip prefix-list plist seq 5 permit 224.1.1.1/32
ip prefix-list plist seq 10 permit 224.1.1.2/32
ip prefix-list plist seq 15 deny 224.1.1.3/32
ip prefix-list plist seq 20 deny 225.0.0.0/8 eq 32

vlan configuration 2
  ip igmp snooping report-policy prefix-list plist interface Ethernet 1/2
  ip igmp snooping report-policy prefix-list plist interface Ethernet 1/3
```

In the above example, the prefix-list permits 224.1.1.1 and 224.1.1.2 but rejects 224.1.1.3 and all the groups in the 225.0.0.0/8 range. The prefix-list is an implicit "deny" if there is no match. If you wish to permit everything else, add **ip prefix-list plist seq 30 permit 224.0.0.0/4 eq 32**.

The following example shows how to configure route maps and use them to filter IGMP snooping reports:

```
route-map rmap permit 10
  match ip multicast group 224.1.1.1/32
route-map rmap permit 20
  match ip multicast group 224.1.1.2/32
route-map rmap deny 30
  match ip multicast group 224.1.1.3/32
route-map rmap deny 40
  match ip multicast group 225.0.0.0/8

vlan configuration 2
  ip igmp snooping report-policy route-map rmap interface Ethernet 1/4
  ip igmp snooping report-policy route-map rmap interface Ethernet 1/5
```

In the above example, the route-map permits 224.1.1.1 and 224.1.1.2 but rejects 224.1.1.3 and all the groups in the 225.0.0.0/8 range. The route-map is an implicit "deny" if there is no match. If you wish to permit everything else, add **route-map rmap permit 50 match ip multicast group 224.0.0.0/4**.

CHAPTER 16

Configuring PIM

This chapter describes how to configure the Protocol Independent Multicast (PIM) features on Cisco NX-OS devices in your IPv4 networks.

About PIM

PIM, which is used between multicast-capable routers, advertises group membership across a routing domain by constructing multicast distribution trees. PIM builds shared distribution trees on which packets from multiple sources are forwarded, as well as source distribution trees on which packets from a single source are forwarded.

Cisco NX-OS supports PIM sparse mode for IPv4 networks (PIM). In PIM sparse mode, multicast traffic is sent only to locations of the network that specifically request it. You can configure PIM to run simultaneously on a router. You can use PIM global parameters to configure rendezvous points (RPs), message packet filtering, and statistics. You can use PIM interface parameters to enable multicast, identify PIM borders, set the PIM hello message interval, and set the designated router (DR) priority.

Note Cisco NX-OS does not support PIM dense mode.

In Cisco NX-OS, multicast is enabled only after you enable the PIM feature on each router and then enable PIM sparse mode on each interface that you want to participate in multicast. You can configure PIM for an IPv4 network . In an IPv4 network, if you have not already enabled IGMP on the router, PIM enables it automatically.

You use the PIM global configuration parameters to configure the range of multicast group addresses to be handled by these distribution modes:

- Any Source Multicast (ASM) provides discovery of multicast sources. It builds a shared tree between sources and receivers of a multicast group and supports switching over to a source tree when a new receiver is added to a group. ASM mode requires that you configure an RP.

For more information about PIM sparse mode and shared distribution trees used by the ASM mode, see RFC 4601.

Note Cisco Nexus® 3550-T does not support the following-

- Cisco Nexus® 3550-T cannot operate as a multicast FHR device.
- Cisco Nexus® 3550-T does not support the formation of source trees (SG-Tree).

Hello Messages

The PIM process begins when the router establishes PIM neighbor adjacencies by sending PIM hello messages to the multicast IPv4 address 224.0.0.13. Hello messages are sent periodically at the interval of 30 seconds. When all neighbors have replied, the PIM software chooses the router with the highest priority in each LAN segment as the designated router (DR). The DR priority is based on a DR priority value in the PIM hello message. If the DR priority value is not supplied by all routers, or the priorities match, the highest IP address is used to elect the DR.

The hello message also contains a hold-time value, which is typically 3.5 times the hello interval. If this hold time expires without a subsequent hello message from its neighbor, the device detects a PIM failure on that link.

The configured hold-time changes may not take effect on first two hellos sent after enabling or disabling PIM on an interface. For the first two hellos sent on the interface, thereafter, the configured hold times will be used. This may cause the PIM neighbor to set the incorrect neighbor timeout value for the initial neighbor setup until a hello with the correct hold time is received.

For added security, you can configure an MD5 hash value that the PIM software uses to authenticate PIM hello messages with PIM neighbors.

Join-Prune Messages

When the DR receives an IGMP membership report message from a receiver for a new group or source, the DR creates a tree to connect the receiver to the source by sending a PIM join message out the interface toward the rendezvous point (ASM mode). The rendezvous point (RP) is the root of a shared tree, which is used by all sources and hosts in the PIM domain in the ASM mode.

When the DR determines that the last host has left a group or source, it sends a PIM prune message to remove the path from the distribution tree.

The routers forward the join or prune action hop by hop up the multicast distribution tree to create (join) or tear down (prune) the path.

Note In this publication, the terms “PIM join message” and “PIM prune message” are used to simplify the action taken when referring to the PIM join-prune message with only a join or prune action.

Join-prune messages are sent as quickly as possible by the software. You can filter the join-prune messages by defining a routing policy.

State Refreshes

PIM requires that multicast entries are refreshed within a 3.5-minute timeout interval. The state refresh ensures that traffic is delivered only to active listeners, and it keeps routers from using unnecessary resources.

To maintain the PIM state, the last-hop DR sends join-prune messages once per minute. State creation applies to (*, G) state as follows:

- (*, G) state creation example—An IGMP (*, G) report triggers the DR to send a (*, G) PIM join message toward the RP.

If the state is not refreshed, the PIM software tears down the distribution tree by removing the forwarding paths in the multicast outgoing interface list of the upstream routers.

Rendezvous Points

A rendezvous point (RP) is a router that you select in a multicast network domain that acts as a shared root for a multicast shared tree. You can configure as many RPs as you like, and you can configure them to cover different group ranges.

Static RP

You can statically configure an RP for a multicast group range. You must configure the address of the RP on every router in the domain.

You can define static RPs for the following reasons:

- To configure routers with the Anycast-RP address
- To manually configure an RP on a device

Note Cisco Nexus® 3550-T only supports and validates Static-RP.

PIM Register Messages

PIM register messages are unicast to the RP by designated routers (DRs) that are directly connected to multicast sources. The PIM register message has the following functions:

- To notify the RP that a source is actively sending to a multicast group.
- To deliver multicast packets sent by the source to the RP for delivery down the shared tree.

The DR continues to send PIM register messages to the RP until it receives a Register-Stop message from the RP. The RP sends a Register-Stop message in either of the following cases:

- The RP has no receivers for the multicast group being transmitted.
- The RP has joined the SPT to the source but has not started receiving traffic from the source.

The PIM triggered register is enabled by default.

You can use the **ip pim register-source** command to configure the IP source address of register messages when the IP source address of a register message is not a uniquely routed address to which the RP can send packets. This situation might occur if the source address is filtered so that the packets sent to it are not forwarded or if the source address is not unique to the network. In these cases, the replies sent from the RP to the source address will fail to reach the DR, resulting in Protocol Independent Multicast sparse mode (PIM-SM) protocol failures.

The following example shows how to configure the IP source address of the register message to the loopback 3 interface of a DR:

```
ip pim register-source loopback 3
```

Note There is no RPF check done for all the multicast routes installed in the Cisco Nexus 3550-T hardware. Any packet hitting the entry is flooded to all the programed receivers, irrespective of the incoming interface.

Note In Cisco NX-OS, PIM register messages are rate limited to avoid overwhelming the RP.

Designated Routers

In PIM ASM mode, the software chooses a designated router (DR) from the routers on each network segment. The DR is responsible for forwarding multicast data for specified groups and sources on that segment.

The DR for each LAN segment is determined as described in the Hello messages.

In ASM mode, the DR is responsible for unicasting PIM register packets to the RP. When a DR receives an IGMP membership report from a directly connected receiver, the shortest path is formed to the RP, which may or may not go through the DR. The result is a shared tree that connects all sources transmitting on the same multicast group to all receivers of that group.

Note Cisco Nexus 3550-T does not forward multicast packets to the designated router if there is no direct receiver connected to the Cisco Nexus 3550-T hardware.

Note PIM-BIDIR mode is not supported in Cisco Nexus 3550-T.

Note ASM Switchover from Shared Tree to Source Tree is not supported in the Cisco Nexus 3550-T 10.1(2t) release.

Prerequisites for PIM

- You are logged onto the device.
- For global commands, you are in the correct virtual routing and forwarding (VRF) mode. The default configuration mode shown in the examples in this chapter applies to the default VRF.

Note Cisco Nexus 3550-T - 10.1(2t) release, supports only default VRF.

Guidelines and Limitations for PIM

PIM has the following guidelines and limitations:

Note In ***Cisco Nexus 3550-T - 10.1(2t) release***, PIM is supported only default VRF.

- Only PIM-ASM mode is supported in the Cisco Nexus® 3550-T switches.
- ***Cisco Nexus 3550-T - 10.1(2t) release*** does not support AutoRP or BSR configuration.
- Below configuration is recommended since {Vrf,S,G} Routes are not supported –
  - Configure **ip pim spt-threshold infinity**
  - Disable PIM-SSM.
  - Even though IGMPv3 snooping is Enabled, {S,G} received from IGMPv3 are not installed in the Cisco Nexus 3550-T - 10.1(2t) release.
- RPF check is not implemented in hardware – hence RPF-failed packet received is forwarded to installed {*,G} route oiflist. Though hardware check is implemented to avoid forward the packet back on incoming L3 port.
- Cisco Nexus® 3550-T switch does cut-through forwarding; hence there is no MTU- check implemented. Hardware buffering is not designed for jumbo packets and packets beyond regular MTU size 1516 is not supported.
- L3 Multicast lookup is not enabled on Trunk ports.
- L3 Multicast has the following scale numbers:
  - L2MCAST - 768 system-wide shared with MAC tabel - {vlan,MAC}

  - EntriesL3MCAST - 384 system-wide {vrf,G,*} Entries only

- L3 multicast result cannot contain trunk ports as an OIF. Any entry computed to install with Trunk-port OIF is not installed in h/w.

- L3 Multicast lookup miss packets are not punted to SUP. Hence, Cisco Nexus® 3550-T switch cannot act as FHR; though if {*,G} tree is already installed it will forward multicast along that path.

- When L3 lookup is done; even the L2 domain multicast receivers receive packets with decremented TTL.

- Cisco Nexus® 3550-T platform switches does not support MSDP.

- For most Cisco Nexus devices, RPF failure traffic is dropped and sent to the CPU at a very low rate to trigger PIM asserts. Cisco Nexus® 3550-T switches, do not check for RPF failure and all traffic is forwarded according to the installed route.

- For first-hop source detection in most Cisco Nexus devices, traffic coming from the first hop is detected based on the source subnet check, and multicast packets are copied to the CPU only if the source belongs to the local subnet. The Cisco Nexus® 3550-T switches do not support FHR function, cannot detect First-hop traffic, hence no multicast packets are sent to the supervisor to learn the local multicast source.

- Cisco NX-OS PIM do not interoperate with any version of PIM dense mode or PIM Sparse Mode version 1.

Guidelines and Limitations for Hello Messages

The following guidelines and limitations apply to Hello Messages:

- Default values for the PIM hello interval are recommended and should not be modified.

Guidelines and Limitations for Rendezvous Points

The following guidelines and limitations apply to Rendezvous Points (RP):

- Cisco Nexus 3550-T - 10.1(2t) release can only operate as a static RP.

Default Settings

This table lists the default settings for PIM parameters.

Table 13: Default PIM Parameters

Parameters	Default
Use shared trees only	Disabled
Flush routes on restart	Disabled
Log neighbor changes	Disabled

Parameters	Default
Auto-RP message action	Disabled **Note** **Do Not Enable** Auto-RP message action since, BSR is not available in Cisco Nexus 3550-T - 10.1(2t) release.
BSR message action	Disabled **Note** **Do not Enable** BSR message action since, BSR is not available in Cisco Nexus 3550-T - 10.1(2t) release.
PIM sparse mode	Disabled
Designated router priority	1
Hello authentication mode	Disabled
Domain border	Disabled **Note** **Do not Enable** since Domain border is not available in Cisco Nexus 3550-T - 10.1(2t) release.

Note Cisco Nexus 3550-T - 10.1(2t) release does not support any policy configuration, hence it is disabled.

Configuring PIM

Note Cisco NX-OS supports only PIM sparse mode version 2. In this publication, “PIM” refers to PIM sparse mode version 2.

You can configure separate ranges of addresses in the PIM domain using the multicast distribution modes described in the table below.

Multicast Distribution Mode	Requires RP Configuration	Description
ASM	Yes	Any source multicast
RPF routes for multicast	No	RPF routes for multicast

Note RPF check is not supported in Cisco Nexus 3550-T - 10.1(2t) release and the Multicast packets are flooded irrespective of RPF failure to programed receivers.

PIM Configuration Tasks

The following steps configure PIM .

Procedure

	Command or Action	Purpose
Step 1	Select the range of multicast groups that you want to configure in each multicast distribution mode.	
Step 2	Enable PIM.	
Step 3	Follow the configuration steps for the multicast distribution modes that you selected in Step 1.	
Step 4		

Enabling the PIM Feature

Before you can access the PIM commands, you must enable the PIM feature.

Before you begin

Ensure that you have installed the Enterprise Services license.

Procedure

	Command or Action	Purpose

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

Step 2	**feature pim** **Example:** `switch(config)# feature pim`	Enables PIM. By default, PIM is disabled.
Step 3	(Optional) **show running-configuration pim** **Example:** `switch(config)# show running-configuration pim`	Shows the running-configuration information for PIM.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring PIM Sparse Mode Parameters

You configure PIM sparse mode on every device interface that you want to participate in a sparse mode domain. You can configure the sparse mode parameters described in the table below.

Table 14: PIM Sparse Mode Parameters

Parameter	Description
Global to the device	
Register rate limit	Configures the IPv4 register rate limit in packets per second. The range is from 1 to 65,535. The default is no limit.
Initial holddown period	Configures the IPv4 initial holddown period in seconds. This holddown period is the time it takes for the MRIB to come up initially. If you want faster convergence, enter a lower value. The range is from 90 to 210. Specify 0 to disable the holddown period. The default is 210.
Per device interface	
PIM sparse mode	Enables PIM on an interface.
Designated router priority	Sets the designated router (DR) priority that is advertised in PIM hello messages on this interface. On a multi-access network with multiple PIM-enabled routers, the router with the highest DR priority is elected as the DR router. If the priorities match, the software elects the DR with the highest IP address. The DR originates PIM register messages for the directly connected multicast sources and sends PIM join messages toward the rendezvous point (RP) for directly connected receivers. Values range from 1 to 4294967295. The default is 1.
Designated router delay	Delays participation in the designated router (DR) election by setting the DR priority that is advertised in PIM hello messages to 0 for a specified period. During this delay, no DR changes occur, and the current switch is given time to learn all of the multicast states on that interface. After the delay period expires, the correct DR priority is sent in the hello packets, which retriggers the DR election. Values range from 3 to 0xffff seconds.

Parameter	Description
Hello authentication mode	Enables an MD5 hash authentication key, or password, in PIM hello messages on the interface so that directly connected neighbors can authenticate each other. The PIM hello messages are IPsec encoded using the Authentication Header (AH) option. You can enter an unencrypted (cleartext) key or one of these values followed by a space and the MD5 authentication key: • 0—Specifies an unencrypted (cleartext) key • 3—Specifies a 3-DES encrypted key • 7—Specifies a Cisco Type 7 encrypted key The authentication key can be up to 16 characters. The default is disabled.
Hello interval	Configures the interval at which hello messages are sent in milliseconds. The range is from 1000 to 18724286. The default is 30000. **Note** See the *Cisco Nexus® 3550-T Verified Scalability Guide* for the verified range of this parameter and associated PIM neighbor scale.

Configuring PIM Sparse Mode Parameters

Procedure

	Command or Action	Purpose

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

Step 2	(Optional) **ip pim register-rate-limit** *rate* **Example:** `switch(config)# ip pim register-rate-limit 1000`	Configures the rate limit in packets per second. The range is from 1 to 65,535. The default is no limit.
Step 3	(Optional) **ip pim spt-threshold infinity group-list** *route-map-name* **Example:** `switch(config)# ip pim spt-threshold infinity group-list my_route-map-name`	Creates the IPv4 PIM (*, G) state only, for the group prefixes defined in the specified route map. Cisco NX-OS Release 3.1 supports up to 1000 route-map entries, and Cisco NX-OS releases prior to 3.1 support up to 500 route-map entries.

	Command or Action	Purpose
		Note The **ip pim use-shared-tree-only group-list** command performs the same function as the **ip pim spt-threshold infinity group-list** command. You can choose to use either command to implement this step.
Step 4	(Optional) [**ip** | **ipv4**] **routing multicast holddown** *holddown-period* **Example:** `switch(config)# ip routing multicast holddown 100`	Configures the initial holddown period in seconds. The range is from 90 to 210. Specify 0 to disable the holddown period. The default is 210.
Step 5	(Optional) **show running-configuration pim** **Example:** `switch(config)# show running-configuration pim`	Displays PIM running-configuration information.

Step 6	**interface** *interface*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 7	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Enables PIM sparse mode on this interface. The default is disabled.
Step 8	(Optional) **ip pim dr-priority** *priority* **Example:** `switch(config-if)# ip pim dr-priority 192`	Sets the designated router (DR) priority that is advertised in PIM hello messages. Values range from 1 to 4294967295. The default is 1.
Step 9	(Optional) **ip pim dr-delay** *delay* **Example:** `switch(config-if)# ip pim dr-delay 3`	Delays participation in the designated router (DR) election by setting the DR priority that is advertised in PIM hello messages to 0 for a specified period. During this delay, no DR changes occur, and the current switch is given time to learn all of the multicast states on that interface. After the delay period expires, the correct DR priority is sent in the hello packets, which retriggers the DR election. Values range from 3 to 0xffff seconds.

	Command or Action	Purpose
		Note This command delays participation in the DR election only upon bootup or following an IP address or interface state change. It is intended for use with multicast-access Layer 3 interfaces only.
Step 10	(Optional) **ip pim hello-authentication ah-md5** *auth-key* **Example:** `switch(config-if)# ip pim hello-authentication ah-md5 my_key`	Enables an MD5 hash authentication key in PIM hello messages. You can enter an unencrypted (cleartext) key or one of these values followed by a space and the MD5 authentication key: • 0—Specifies an unencrypted (cleartext) key • 3—Specifies a 3-DES encrypted key • 7—Specifies a Cisco Type 7 encrypted key The key can be up to 16 characters. The default is disabled.
Step 11	(Optional) **ip pim hello-interval** *interval* **Example:** `switch(config-if)# ip pim hello-interval 25000`	Configures the interval at which hello messages are sent in milliseconds. The range is from 1000 to 18724286. The default is 30000. **Note** The minimum value is 1 millisecond.
Step 12	(Optional) **show ip pim interface** [*interface* | **brief**] **Example:** `switch(config-if)# show ip pim interface`	Displays PIM interface information.
Step 13	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring ASM

To configure ASM mode, you configure sparse mode and the RP selection method, where you indicate the distribution mode and assign the range of multicast groups.

Configuring Static RPs

You can configure an RP statically by configuring the RP address on every router that will participate in the PIM domain.

Note We recommend that the RP address uses the loopback interface and also the interface with the RP address must have **ip pim sparse-mode** enabled.

Configuring Static RPs

Before you begin

Ensure that you have installed the Enterprise Services license and enabled PIM.

Procedure

	Command or Action	Purpose

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

Step 2	**ip pim rp-address** *rp-address* **Example:** `switch(config)# ip pim rp-address 192.0.2.33`	Configures a PIM static RP address for a multicast groups. You can specify a prefix-list policy name for the static RP address. The mode is ASM.
Step 3	(Optional) **show ip pim group-range** [*ip-prefix*] **Example:** `switch(config)# show ip pim group-range`	Displays PIM RP information.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Shared Trees Only for ASM

You can configure shared trees only on the last-hop router for Any Source Multicast (ASM) groups, which means that the router never switches over from the shared tree to the SPT when a receiver joins an active group.

Note The Cisco Nexus® 3550-T supports shared-tree feature only.

Configuring Message Filtering

You can configure filtering of the PIM messages described in the table below.

Table 15: PIM Message Filtering

Message Type	Description
Global to the Device	
Log Neighbor changes	Enables syslog messages that list the neighbor state changes to be generated. The default is disabled.

Configuring Message Filtering

Before you begin

Ensure that you have installed the Enterprise Services license and enabled PIM.

Procedure

	Command or Action	Purpose

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

Step 2	(Optional) **ip pim log-neighbor-changes**
	Example:
	`switch(config)# ip pim`
	`log-neighbor-changes`
		Enables syslog messages that list the neighbor state changes to be generated. The default is disabled.

Step 3	(Optional) **show run pim** **Example:** `switch(config-if)# show run pim`	Displays PIM configuration commands.

Step 4	(Optional) **copy running-config startup-config**
	Example:
	`switch(config-if)# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Restarting the PIM Processes

When routes are flushed, they are removed from the Multicast Routing Information Base (MRIB) and the Multicast Forwarding Information Base (MFIB).

When you restart PIM, the following tasks are performed:

- The PIM database is deleted.
- The MRIB and MFIB are unaffected and forwarding of traffic continues.
- The multicast route ownership is verified through the MRIB.
- Periodic PIM join and prune messages from neighbors are used to repopulate the database.

Restarting the PIM Process

Before you begin

Ensure that you have installed the Enterprise Services license and enabled PIM.

Procedure

	Command or Action	Purpose
Step 1	**restart pim** **Example:** `switch# restart pim`	Restarts the PIM process. **Note** Traffic loss might occur during the restart process.

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

Step 3	**ip pim flush-routes** **Example:** `switch(config)# ip pim flush-routes`	Removes routes when the PIM process is restarted. By default, routes are not flushed.
Step 4	(Optional) **show running-configuration pim** **Example:** `switch(config)# show running-configuration pim`	Displays the PIM running-configuration information, including the **flush-routes** command.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the PIM Configuration

To display the PIM configuration information, perform one of the following tasks.

Command	Description
show ip mroute [*ip-address*] [**detail** | **summary**]	Displays the IP multicast routing table. The **detail** option displays detailed route attributes. The **summary** option displays route counts and packet rates. **Note** This command also displays multicast counters for Cisco Nexus® 3550-T switches, if the multicast heavy template is enabled. See sample outputs below.
show ip pim group-range [*ip-prefix*]	Displays the learned or configured group ranges and modes. For similar information, see the **show ip pim rp** command.
show ip pim interface [*interface* | **brief**]	Displays information by the interface.
show ip pim neighbor [**interface** *interface* | *ip-prefix*]	Displays neighbors by the interface.
show ip pim oif-list *group* [*source*]	Displays all the interfaces in the outgoing interface (OIF) list.
show ip pim route [*source* | *group* [*source*]]	Displays information for each multicast route, including interfaces on which a PIM join for that (*, G) has been received.
show ip pim rp [*ip-prefix*]	Displays rendezvous points (RPs) known to the software, how they were learned, and their group ranges. For similar information, see the **show ip pim group-range** command.
show running-config pim	Displays the running-configuration information.
show startup-config pim	Displays the startup-configuration information.
show ip pim [**detail**]	Displays pim details information.

Displaying Statistics

You can display and clear PIM statistics by using the commands in this section.

Displaying PIM Statistics

You can display the PIM statistics and memory usage using these commands.

Command	Description
show ip pim policy statistics	Displays policy statistics for register, RP, and join-prune message policies.
show ip pim statistics	Displays global statistics.

Clearing PIM Statistics

You can clear the PIM statistics using these commands.

Command	Description
clear ippim interface statistics *interface*	Clears counters for the specified interface.
clear ip pim policy statistics	Clears policy counters for register, RP, and join-prune message policies.
clear ip pim statistics	Clears global counters handled by the PIM process.

Related Documents

Related Topic	Document Title

Standards

MIBs

MIBs	MIBs Link
MIBs related to PIM	To locate and download supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

PART V

Cisco Nexus 3550-T Unicast Routing Configuration Guide

CHAPTER 17

Unicast Routing Overview

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Information About Layer 3 Unicast Routing

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

Routing Fundamentals

Routing protocols use a metric to evaluate the best path to the destination. A metric is a standard of measurement, such as a path bandwidth, that routing algorithms use to determine the optimal path to a destination. To aid path determination, routing algorithms initialize and maintain routing tables that contain route information such as the IP destination address, the address of the next router, or the next hop. Destination and next-hop associations tell a router that an IP destination can be reached optimally by sending the packet to a particular router that represents the next hop on the way to the final destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with the next hop. See the *Unicast RIB* section for more information about the route table.

Routing tables can contain other information, such as the data about the desirability of a path. Routers compare metrics to determine optimal routes, and these metrics differ depending on the design of the routing algorithm used. See the *Routing Metrics* section.

Routers communicate with one another and maintain their routing tables by transmitting a variety of messages. The routing update message is one such message that consists of all or a portion of a routing table. By analyzing routing updates from all other routers, a router can build a detailed picture of the network topology. A link-state advertisement, which is another example of a message sent between routers, informs other routers of the link state of the sending router. You can also use link information to enable routers to determine optimal routes to network destinations. For more information, see the *Routing Algorithms* section.

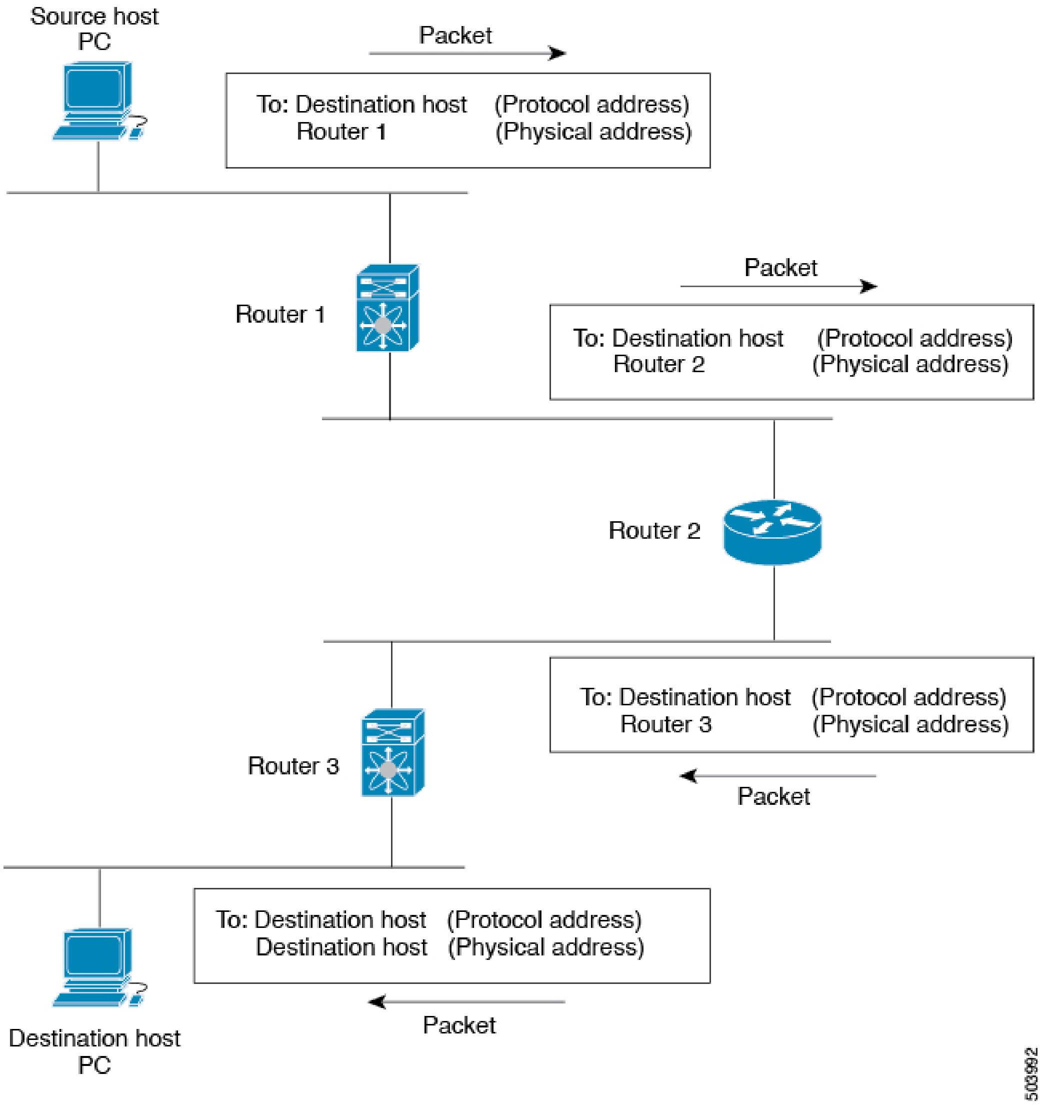
Packet Switching

In packet switching, a host determines that it must send a packet to another host. Having acquired a router address by some means, the source host sends a packet that is addressed specifically to the router physical (Media Access Control [MAC]-layer) address but with the IP (network layer) address of the destination host.

The router examines the destination IP address and tries to find the IP address in the routing table. If the router does not know how to forward the packet, it typically drops the packet. If the router knows how to forward the packet, it changes the destination MAC address to the MAC address of the next-hop router and transmits the packet.

The next hop might be the ultimate destination host or another router that executes the same switching decision process. As the packet moves through the internetwork, its physical address changes, but its protocol address remains constant (see the following figure).

Figure 8: Packet Header Updates Through a Network


Source host
PC
Packet
To: Destination host (Protocol address)
Router 1 (Physical address)
Router 1
Packet
To: Destination host (Protocol address)
Router 2 (Physical address)
Router 2
Router 3
To: Destination host (Protocol address)
Router 3 (Physical address)
Packet
To: Destination host (Protocol address)
Destination host (Physical address)
Packet
Destination host
PC
503992

Routing Metrics

Routing algorithms use many different metrics to determine the best route. Sophisticated routing algorithms can base route selection on multiple metrics.

Path Length

The path length is the most common routing metric. Some routing protocols allow you to assign arbitrary costs to each network link. In this case, the path length is the sum of the costs associated with each link traversed. Other routing protocols define the hop count, which is a metric that specifies the number of passes through internetworking products, such as routers, that a packet must take from a source to a destination.

Reliability

The reliability, in the context of routing algorithms, is the dependability (in terms of the bit-error rate) of each network link. Some network links might go down more often than others. After a network fails, certain network links might be repaired more easily or more quickly than other links. The reliability factors that you can take into account when assigning the reliability rating are arbitrary numeric values that you usually assign to network links.

Routing Delay

The routing delay is the length of time required to move a packet from a source to a destination through the internetwork. The delay depends on many factors, including the bandwidth of intermediate network links, the port queues at each router along the way, the network congestion on all intermediate network links, and the physical distance that the packet must travel. Because the routing delay is a combination of several important variables, it is a common and useful metric.

Bandwidth

The bandwidth is the available traffic capacity of a link. For example, a 10-Gigabit Ethernet link is preferable to a 1-Gigabit Ethernet link. Although the bandwidth is the maximum attainable throughput on a link, routes through links with greater bandwidth do not necessarily provide better routes than routes through slower links. For example, if a faster link is busier, the actual time required to send a packet to the destination could be greater.

Load

The load is the degree to which a network resource, such as a router, is busy. You can calculate the load in a variety of ways, including CPU usage and packets processed per second. Monitoring these parameters on a continual basis can be resource intensive.

Communication Cost

The communication cost is a measure of the operating cost to route over a link. The communication cost is another important metric, especially if you do not care about performance as much as operating expenditures. For example, the line delay for a private line might be longer than a public line, but you can send packets over your private line rather than through the public lines that cost money for usage time.

Router IDs

Each routing process has an associated router ID. You can configure the router ID to any interface in the system. If you do not configure the router ID, Cisco NX-OS selects the router ID based on the following criteria:

- Cisco NX-OS prefers loopback0 over any other interface. If loopback0 does not exist, then Cisco NX-OS prefers the first loopback interface over any other interface type.
- If you have not configured a loopback interface, Cisco NX-OS uses the first interface in the configuration file as the router ID. If you configure any loopback interface after Cisco NX-OS selects the router ID, the loopback interface becomes the router ID. If the loopback interface is not loopback0 and you configure loopback0 with an IP address, the router ID changes to the IP address of loopback0.
- If the interface that the router ID is based on changes, that new IP address becomes the router ID. If any other interface changes its IP address, there is no router ID change.

Convergence

A key aspect to measure for any routing algorithm is how much time a router takes to react to network topology changes. When a part of the network changes for any reason, such as a link failure, the routing information in different routers might not match. Some routers will have updated information about the changed topology, while other routers will still have the old information. The convergence is the amount of time before all routers in the network have updated, matching routing information. The convergence time varies depending on the routing algorithm. Fast convergence minimizes the chance of lost packets caused by inaccurate routing information.

Route Redistribution

If you have multiple routing protocols configured in your network, you can configure these protocols to share routing information by configuring route redistribution in each protocol. For example, you can configure the Open Shortest Path First (OSPF) protocol to advertise routes learned from the Border Gateway Protocol (BGP). You can also redistribute static routes into any dynamic routing protocol. The router that is redistributing routes from another protocol sets a fixed route metric for those redistributed routes, which prevents incompatible route metrics between the different routing protocols. For example, routes redistributed from EIGRP into OSPF are assigned a fixed link cost metric that OSPF understands.

Note You are required to use route maps when you configure the redistribution of routing information.

Route redistribution also uses an administrative distance (see see the *Administrative Distance* section) to distinguish between routes learned from two different routing protocols. The preferred routing protocol is given a lower administrative distance so that its routes are picked over routes from another protocol with a higher administrative distance assigned.

Administrative Distance

An administrative distance is a rating of the trustworthiness of a routing information source. A higher value indicates a lower trust rating. Typically, a route can be learned through more than one protocol. Administrative distance is used to discriminate between routes learned from more than one protocol. The route with the lowest administrative distance is installed in the IP routing table.

Stub Routing

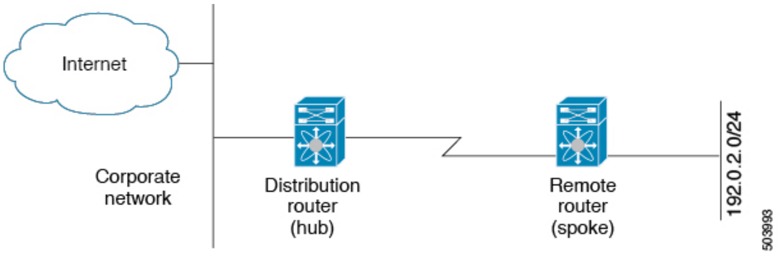
You can use stub routing in a hub-and-spoke network topology, where one or more end (stub) networks are connected to a remote router (the spoke) that is connected to one or more distribution routers (the hub). The remote router is adjacent only to one or more distribution routers. The only route for IP traffic to follow into the remote router is through a distribution router. This type of configuration is commonly used in WAN topologies in which the distribution router is directly connected to a WAN. The distribution router can be connected to many more remote routers. Often, the distribution router is connected to 100 or more remote routers. In a hub-and-spoke topology, the remote router must forward all nonlocal traffic to a distribution router, so it becomes unnecessary for the remote router to hold a complete routing table. Generally, the distribution router sends only a default route to the remote router.

Only specified routes are propagated from the remote (stub) router. The stub router responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message “inaccessible.” A router that is configured as a stub sends a special peer information packet to all neighboring routers to report its status as a stub router.

Any neighbor that receives a packet that informs it of the stub status does not query the stub router for any routes, and a router that has a stub peer does not query that peer. The stub router depends on the distribution router to send the proper updates to all peers.

The following figure shows a simple hub-and-spoke configuration.

Figure 9: Simple Hub-and-Spoke Network



Stub routing does not prevent routes from being advertised to the remote router. The figure **Simple Hub-and-Spoke Network** shows that the remote router can access the corporate network and the Internet through the distribution router only. A full route table on the remote router, in this example, serves no functional purpose because the path to the corporate network and the Internet is always through the distribution router. A larger route table reduces only the amount of memory required by the remote router. The bandwidth and memory used can be lessened by summarizing and filtering routes in the distribution router. In this network topology, the remote router does not need to receive routes that have been learned from other networks because the remote router must send all nonlocal traffic, regardless of its destination, to the distribution router. To configure a true stub network, you should configure the distribution router to send only a default route to the remote router.

OSPF supports stub areas, and the Enhanced Interior Gateway Routing Protocol (EIGRP) supports stub routers.

Note The EIGRP stub routing feature should be used only on stub devices. A stub device is defined as a device connected to the network core or distribution layer through which core transit traffic should not flow. The only route for IP traffic to follow into the remote router is through a distribution router. A stub device should not have any EIGRP neighbors other than distribution devices. Ignoring this restriction will cause undesirable behavior.

Routing Algorithms

Routing algorithms determine how a router gathers and reports reachability information, how it deals with topology changes, and how it determines the optimal route to a destination. Various types of routing algorithms exist, and each algorithm has a different impact on network and router resources. Routing algorithms use a variety of metrics that affect calculation of optimal routes. You can classify routing algorithms by type, such as static or dynamic, and interior or exterior.

Static Routes and Dynamic Routing Protocols

Static routes are route table entries that you manually configure. These static routes do not change unless you reconfigure them. Static routes are simple to design and work well in environments where network traffic is relatively predictable and where network design is relatively simple.

Because static routing systems cannot react to network changes, you should not use them for large, constantly changing networks. Most routing protocols today use dynamic routing algorithms that adjust to changing network circumstances by analyzing incoming routing update messages. If the message indicates that a network change has occurred, the routing software recalculates routes and sends out new routing update messages. These messages permeate the network, triggering routers to rerun their algorithms and change their routing tables accordingly.

You can supplement dynamic routing algorithms with static routes where appropriate. For example, you should configure each subnetwork with a static route to the IP default gateway or router of last resort (a router to which all unrouteable packets are sent).

Interior and Exterior Gateway Protocols

You can separate networks into unique routing domains or autonomous systems. An autonomous system is a portion of an internetwork under common administrative authority that is regulated by a particular set of administrative guidelines. Routing protocols that route between autonomous systems are called exterior gateway protocols or interdomain protocols. The Border Gateway Protocol (BGP) is an example of an exterior gateway protocol. Routing protocols used within an autonomous system are called interior gateway protocols or intradomain protocols. EIGRP and OSPF are examples of interior gateway protocols.

Distance Vector Protocols

Distance vector protocols use distance vector algorithms (also known as Bellman-Ford algorithms) that call for each router to send all or some portion of its routing table to its neighbors. Distance vector algorithms define routes by distance (for example, the number of hops to the destination) and direction (for example, the next-hop router). These routes are then broadcast to the directly connected neighbor routers. Each router uses these updates to verify and update the routing tables.

To prevent routing loops, most distance vector algorithms use split horizon with poison reverse which means that the routes learned from an interface are set as unreachable and advertised back along the interface that they were learned on during the next periodic update. This process prevents the router from seeing its own route updates coming back.

Distance vector algorithms send updates at fixed intervals but can also send updates in response to changes in route metric values. These triggered updates can speed up the route convergence time. The Routing Information Protocol (RIP) is a distance vector protocol.

Link-State Protocols

The link-state protocols, also known as shortest path first (SPF), share information with neighboring routers. Each router builds a link-state advertisement (LSA) that contains information about each link and directly connected neighbor router.

Each LSA has a sequence number. When a router receives an LSA and updates its link-state database, the LSA is flooded to all adjacent neighbors. If a router receives two LSAs with the same sequence number (from the same router), the router does not flood the last LSA that it received to its neighbors because it wants to prevent an LSA update loop. Because the router floods the LSAs immediately after it receives them, the convergence time for link-state protocols is minimized.

Discovering neighbors and establishing adjacency is an important part of a link state protocol. Neighbors are discovered using special Hello packets that also serve as keepalive notifications to each neighbor router. Adjacency is the establishment of a common set of operating parameters for the link-state protocol between neighbor routers.

The LSAs received by a router are added to the router's link-state database. Each entry consists of the following parameters:

- Router ID (for the router that originated the LSA)
- Neighbor ID
- Link cost
- Sequence number of the LSA
- Age of the LSA entry

The router runs the SPF algorithm on the link-state database, building the shortest path tree for that router. This SPF tree is used to populate the routing table.

In link-state algorithms, each router builds a picture of the entire network in its routing tables. The link-state algorithms send small updates everywhere, while distance vector algorithms send larger updates only to neighboring routers.

Because they converge more quickly, link-state algorithms are less likely to cause routing loops than distance vector algorithms. However, link-state algorithms require more CPU power and memory than distance vector algorithms and they can be more expensive to implement and support. Link-state protocols are generally more scalable than distance vector protocols.

OSPF is an example of a link-state protocol.

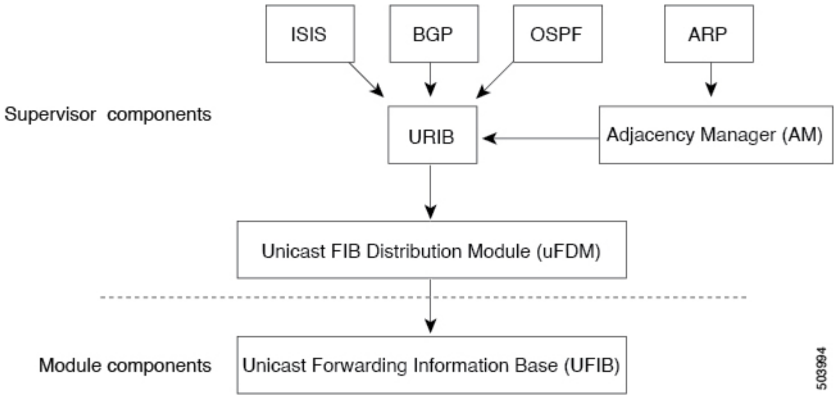
Cisco NX-OS Forwarding Architecture

The Cisco NX-OS forwarding architecture is responsible for processing all routing updates and populating the forwarding information to all modules in the chassis.

Unicast RIB

The Cisco NX-OS forwarding architecture consists of multiple components, as shown in the following figure.

Figure 10: Cisco NX-OS Forwarding Architecture



The unicast RIB exists on the active supervisor. It maintains the routing table with directly connected routes, static routes, and routes learned from dynamic unicast routing protocols. The unicast RIB also collects adjacency information from sources such as the Address Resolution Protocol (ARP). The unicast RIB determines the best next hop for a given route and populates the FIB by using the services of the unicast FIB Distribution Module (FDM).

Each dynamic routing protocol must update the unicast RIB for any route that has timed out. The unicast RIB then deletes that route and recalculates the best next hop for that route (if an alternate path is available).

Adjacency Manager

The adjacency manager exists on the active supervisor and maintains adjacency information for different protocols including ARP, Neighbor Discovery Protocol (NDP), and static configuration. The most basic adjacency information is the Layer 3 to Layer 2 address mapping discovered by these protocols. Outgoing Layer 2 packets use the adjacency information to complete the Layer 2 header.

The adjacency manager can trigger ARP requests to find a particular Layer 3 to Layer 2 mapping. The new mapping becomes available when the corresponding ARP reply is received and processed.

Unicast Forwarding Distribution Module

The unicast Forwarding Distribution Module (FDM) exists on the active supervisor and distributes the forwarding path information from the unicast RIB and other sources. The unicast RIB generates forwarding information that the unicast FIB programs into the hardware forwarding tables on the standby supervisor and the modules. The unicast FDM also downloads the FIB information to newly inserted modules.

The unicast FDM gathers adjacency information, rewrite information, and other platform-dependent information when updating routes in the unicast FIB. The adjacency and rewrite information consists of interface, next hop, and Layer 3 to Layer 2 mapping information. The interface and next-hop information is received in route updates from the unicast RIB. The Layer 3 to Layer 2 mapping is received from the adjacency manager.

FIB

The unicast FIB exists on supervisors and switching modules and builds the information used for the hardware forwarding engine. The unicast FIB receives route updates from the unicast FDM and sends the information to be programmed in the hardware forwarding engine. The unicast FIB controls the addition, deletion, and modification of routes, paths, and adjacencies.

Based on route update messages, the unicast FIB maintains a per-VRF prefix and next-hop adjacency information database. The next-hop adjacency data structure contains the next-hop IP address and the Layer 2 rewrite information. Multiple prefixes could share a next-hop adjacency information structure.

Hardware Forwarding

Cisco NX-OS supports distributed packet forwarding. The ingress port takes relevant information from the packet header and passes the information to the local switching engine. The local switching engine does the Layer 3 lookup and uses this information to rewrite the packet header. The ingress module forwards the packet to the egress port. If the egress port is on a different module, the packet is forwarded using the switch fabric to the egress module. The egress module does not participate in the Layer 3 forwarding decision.

You also use the **show platform fib** or **show platform forwarding** commands to display details on hardware forwarding.

Software Forwarding

The software forwarding path in Cisco NX-OS is used mainly to handle features that are not supported in the hardware or to handle errors encountered during the hardware processing. Typically, packets with IP options or packets that need fragmentation are passed to the CPU on the active supervisor. All packets that should be switched in the software or terminated go to the supervisor. The supervisor uses the information provided by the unicast RIB and the adjacency manager to make the forwarding decisions. The module is not involved in the software forwarding path.

Software forwarding is controlled by control plane policies and rate limiters. For more information, see the Cisco NX-OS Security Configuration Guide.

Summary of Layer 3 Unicast Routing Features

This section provides a brief introduction to the Layer 3 unicast features and protocols supported in Cisco NX-OS.

IPv4

Layer 3 uses either the IPv4 protocol. For more information, see the *Configuring IPV4* section.

OSPF

The Open Shortest Path First (OSPF) protocol is a link-state routing protocol used to exchange network reachability information within an autonomous system. Each OSPF router advertises information about its active links to its neighbor routers. Link information consists of the link type, the link metric, and the neighbor router that is connected to the link. The advertisements that contain this link information are called link-state advertisements. For more information, see the *Configuring OSPFv2* section.

BGP

The Border Gateway Protocol (BGP) is an inter-autonomous system routing protocol. A BGP router advertises network reachability information to other BGP routers using Transmission Control Protocol (TCP) as its reliable transport mechanism. The network reachability information includes the destination network prefix, a list of autonomous systems that needs to be traversed to reach the destination, and the next-hop router. Reachability information contains additional path attributes such as preference to a route, origin of the route, community and others. For more information, see the *Configuring Basic BGP* and *Configuring Advanced BGP* sections.

Static Routing

Static routing allows you to enter a fixed route to a destination. This feature is useful for small networks where the topology is simple. Static routing is also used with other routing protocols to control default routes and route distribution. For more information, see the *Configuring Static Routing* section.

First Hop Redundancy Protocols

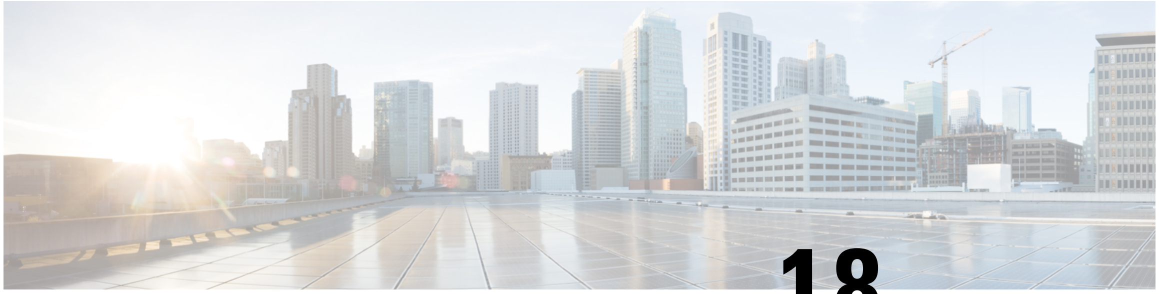
First hop redundancy protocols (FHRP), such as the Virtual Router Redundancy Protocol (VRRP), allow you to provide redundant connections to your hosts. If an active first-hop router fails, the FHRP automatically selects a standby router to take over. You do not need to update the hosts with new IP addresses because the address is virtual and shared between each router in the FHRP group. For more information on VRRP, see the *Configuring VRRP* section.

Object Tracking

Object tracking allows you to track specific objects on the network, such as the interface line protocol state, IP routing, and route reachability, and take action when the tracked object's state changes. This feature allows you to increase the availability of the network and shorten the recovery time if an object state goes down.

Related Topics

Feature Name	Feature Information
Layer 3 features	*Cisco Nexus® 3550-T Multicast Routing Configuration* section *Cisco Cisco NX-OS Series NX-OS High Availability and Redundancy Guide* Exploring Autonomous System Numbers: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_9-1/autonomous_system_numbers.html



CHAPTER 18

Configuring IPv4

This chapter describes how to configure Internet Protocol version 4 (IPv4), which includes addressing, Address Resolution Protocol (ARP), and Internet Control Message Protocol (ICMP), on the Cisco NX-OS device.

This chapter includes the following sections:

About IPv4

You can configure IP on the device to assign IP addresses to network interfaces. When you assign IP addresses, you enable the interfaces and allow communication with the hosts on those interfaces.

You can configure an IP address as primary or secondary on a device. An interface can have one primary IP address and multiple secondary addresses. All networking devices on an interface should share the same primary IP address because the packets that are generated by the device always use the primary IPv4 address. Each IPv4 packet is based on the information from a source or destination IP address. For more information, see the Multiple IPv4 Addresses, on page 212 section.

You can use a subnet to mask the IP addresses. A mask is used to determine what subnet an IP address belongs to. An IP address contains the network address and the host address. A mask identifies the bits that denote the network number in an IP address. When you use the mask to subnet a network, the mask is then referred to as a subnet mask. Subnet masks are 32-bit values that allow the recipient of IP packets to distinguish the network ID portion of the IP address from the host ID portion of the IP address.

The IP feature is responsible for handling IPv4 packets that terminate in the supervisor module, as well as forwarding of IPv4 packets, which includes IPv4 unicast route lookup and software access control list (ACL) forwarding. The IP feature also manages the network interface IP address configuration, duplicate address checks, static routes, and packet send/receive interface for IP clients.

Multiple IPv4 Addresses

Cisco NX-OS supports multiple IP addresses per interface. You can specify an unlimited number of secondary addresses for a variety of situations. The most common are as follows:

- When there are not enough host IP addresses for a particular network interface. For example, if your subnetting allows up to 254 hosts per logical subnet, but on one physical subnet you must have 300 host addresses, then you can use secondary IP addresses on the routers or access servers to allow you to have two logical subnets that use one physical subnet.
- Two subnets of a single network might otherwise be separated by another network. You can create a single network from subnets that are physically separated by another network by using a secondary address. In these instances, the first network is extended, or layered on top of the second network. A subnet cannot appear on more than one active interface of the router at a time.

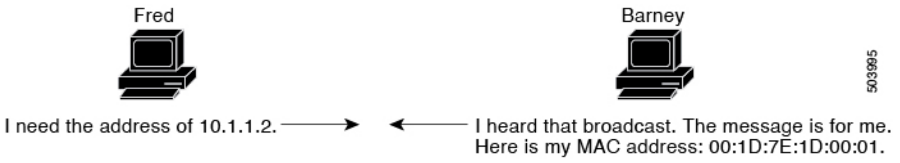
Note If any device on a network segment uses a secondary IPv4 address, all other devices on that same network interface must also use a secondary address from the same network or subnet. The inconsistent use of secondary addresses on a network segment can quickly cause routing loops.

Address Resolution Protocol

Networking devices and Layer 3 switches use Address Resolution Protocol (ARP) to map IP (network layer) addresses to (Media Access Control [MAC]-layer) addresses to enable IP packets to be sent across networks. Before a device sends a packet to another device, it looks in its own ARP cache to see if there is a MAC address and corresponding IP address for the destination device. If there is no entry, the source device sends a broadcast message to every device on the network.

Each device compares the IP address to its own. Only the device with the matching IP address replies to the device that sends the data with a packet that contains the MAC address for the device. The source device adds the destination device MAC address to its ARP table for future reference, creates a data-link header and trailer that encapsulates the packet, and proceeds to transfer the data. The following figure shows the ARP broadcast and response process.

Figure 11: ARP Process


Fred
Barney
503995
I need the address of 10.1.1.2.
I heard that broadcast. The message is for me.
Here is my MAC address: 00:1D:7E:1D:00:01.

When the destination device lies on a remote network that is beyond another device, the process is the same except that the device that sends the data sends an ARP request for the MAC address of the default gateway. After the address is resolved and the default gateway receives the packet, the default gateway broadcasts the destination IP address over the networks connected to it. The device on the destination device network uses ARP to obtain the MAC address of the destination device and delivers the packet. ARP is enabled by default.

ARP Caching

ARP caching minimizes broadcasts and limits wasteful use of network resources. The mapping of IP addresses to MAC addresses occurs at each hop (device) on the network for every packet sent over an internetwork, which may affect network performance.

ARP caching stores network addresses and the associated data-link addresses in the memory for a period of time, which minimizes the use of valuable network resources to broadcast for the same address each time that a packet is sent. You must maintain the cache entries that are set to expire periodically because the information might become outdated. Every device on a network updates its tables as addresses are broadcast.

Static and Dynamic Entries in the ARP Cache

Static routing requires that you manually configure the IP addresses, subnet masks, gateways, and corresponding MAC addresses for each interface of each device. Static routing requires more work to maintain the route table. You must update the table each time you add or change routes.

Dynamic routing uses protocols that enable the devices in a network to exchange routing table information with each other. Dynamic routing is more efficient than static routing because the route table is automatically updated unless you add a time limit to the cache. The default time limit is 25 minutes but you can modify the time limit if the network has many routes that are added and deleted from the cache.

Devices That Do Not Use ARP

When a network is divided into two segments, a bridge joins the segments and filters traffic to each segment based on MAC addresses. The bridge builds its own address table, which uses MAC addresses only. A device has an ARP cache that contains both IP addresses and the corresponding MAC addresses.

Passive hubs are central-connection devices that physically connect other devices in a network. They send messages out on all their ports to the devices and operate at Layer 1 but do not maintain an address table.

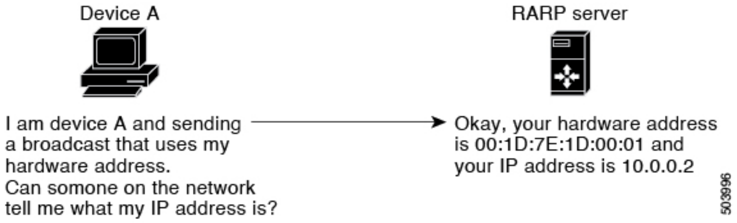
Layer 2 switches determine which port of a device receives a message that is sent only to that port. However, Layer 3 switches are devices that build an ARP cache (table).

Reverse ARP

Reverse ARP (RARP) as defined by RFC 903 works the same way as ARP, except that the RARP request packet requests an IP address instead of a MAC address. RARP often is used by diskless workstations because this type of device has no way to store IP addresses to use when they boot. The only address that is known is the MAC address because it is burned into the hardware.

Use of RARP requires an RARP server on the same network segment as the router interface. The following figure shows how RARP works.

Figure 12: Reverse ARP


Device A
RARP server
I am device A and sending a broadcast that uses my hardware address. Can somone on the network tell me what my IP address is?
Okay, your hardware address is 00:1D:7E:1D:00:01 and your IP address is 10.0.0.2
503996

RARP has several limitations. Because of these limitations, most businesses use Dynamic Host Control Protocol (DHCP) to assign IP addresses dynamically. DHCP is cost effective and requires less maintenance than RARP. The following are the most important limitations:

- Because RARP uses hardware addresses, if the internetwork is large with many physical networks, a RARP server must be on every segment with an additional server for redundancy. maintaining two servers for every segment is costly.
- Each server must be configured with a table of static mappings between the hardware addresses and IP addresses. Maintenance of the IP addresses is difficult.
- RARP only provides IP addresses of the hosts and not subnet masks or default gateways.

Proxy ARP

Proxy ARP enables a device that is physically located on one network appear to be logically part of a different physical network connected to the same device or firewall. Proxy ARP allows you to hide a device with a public IP address on a private network behind a router and still have the device appear to be on the public network in front of the router. By hiding its identity, the router accepts responsibility for routing packets to the real destination. Proxy ARP can help devices on a subnet reach remote subnets without configuring routing or a default gateway.

When devices are not in the same data link layer network but in the same IP network, they try to transmit data to each other as if they are on the local network. However, the router that separates the devices does not send a broadcast message because routers do not pass hardware-layer broadcasts and the addresses cannot be resolved.

When you enable proxy ARP on the device and it receives an ARP request, it identifies the request as a request for a system that is not on the local LAN. The device responds as if it is the remote destination for which the broadcast is addressed, with an ARP response that associates the device’s MAC address with the remote destination's IP address. The local device believes that it is directly connected to the destination, while in reality its packets are being forwarded from the local subnetwork toward the destination subnetwork by their local device. By default, proxy ARP is disabled.

Local Proxy ARP

You can use local proxy ARP to enable a device to respond to ARP requests for IP addresses within a subnet where normally no routing is required. When you enable local proxy ARP, ARP responds to all ARP requests for IP addresses within the subnet and forwards all traffic between hosts in the subnet. Use this feature only on subnets where hosts are intentionally prevented from communicating directly by the configuration on the device to which they are connected.

Gratuitous ARP

Gratuitous ARP sends a request with an identical source IP address and a destination IP address to detect duplicate IP addresses. Cisco NX-OS supports enabling or disabling gratuitous ARP requests or ARP cache updates.

ICMP

You can use the Internet Control Message Protocol (ICMP) to provide message packets that report errors and other information that is relevant to IP processing. ICMP generates error messages, such as ICMP destination unreachable messages, ICMP Echo Requests (which send a packet on a round trip between two hosts) and Echo Reply messages. ICMP also provides many diagnostic functions and can send and redirect error packets to the host. By default, ICMP is enabled.

Some of the ICMP message types are as follows:

- Network error messages
- Network congestion messages
- Troubleshooting information
- Timeout announcements

Note ICMP redirects are disabled on interfaces where the local proxy ARP feature is enabled.

Prerequisites for IPv4

IPv4 has the following prerequisites:

- IPv4 can only be configured on Layer 3 interfaces.

Guidelines and Limitations for IPv4

IPv4 has the following configuration guidelines and limitations:

- You can configure a secondary IP address only after you configure the primary IP address.
- ***Cisco Nexus 3550-T - 10.1(2t) release*** switch does not support hardware load balancing across IPv4 paths and installs only first path from an IPv4 ECMP in hardware. The additional paths are only available in software routing table and next one is updated to hardware when first one goes down. Additionally, there is a syslog generated when a ECMP path is computed to install in hardware.

Parameters	**Scale Numbers**
IP-Host-Route	3072 (max) (per Quad)
L3 ARP/Adjacencies	386

Parameters	Scale Numbers
IP-Routes	2304 (max) (per Quad) **Note** Not all IPv4 route distribution can fit in the Cisco Nexus® 3550-T hardware. It does software forwarding if the route cannot fit in the hardware tables.

Default Settings

The table below lists the default settings for IP parameters.

Parameters	Default
ARP timeout	1500 seconds
Proxy ARP	Disabled

Configuring IPv4

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Configuring IPv4 Addressing

You can assign a primary IP address for a network interface.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *number*
	Example:
	`switch(config)# interface ethernet 1/3`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**ip address** *ip-address/length* [*secondary*] **Example:**	Specifies a primary or secondary IPv4 address for an interface.

	Command or Action	Purpose
	`switch(config-if)# ip address 192.2.1.1 255.0.0.0`	• The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means the corresponding address bit belongs to the network address. • The network mask can be indicated as a slash (/) and a number, which is the prefix length. The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash must precede the decimal value and there must be no space between the IP address and the slash.
Step 4	(Optional) **show ip interface** **Example:** `switch(config-if)# show ip interface`	Displays interfaces configured for IPv4.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Multiple IP Addresses

You can only add secondary IP addresses after you configure primary IP addresses.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *number*
	Example:
	`switch(config)# interface ethernet 1/3`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**ip address** *ip-address/length* [*secondary*] **Example:** `switch(config-if)# ip address 192.168.1.1 255.0.0.0 secondary`	Specifies a the configured address as a secondary IPv4 address.

	Command or Action	Purpose
Step 4	(Optional) **show ip interface** **Example:** `switch(config-if)# show ip interface`	Displays interfaces configured for IPv4.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Saves this configuration change. **Note** Cisco Nexus® 3550-T switch does not support hardware load balancing across IPv4 paths and installs only first path from an IPv4 ECMP in hardware. The additional paths are only available in software routing table and next one is updated to hardware when first one goes down. **Note** Cisco Nexus® 3550-T switch forwards L3 packets with MyMac as destination according to route-lookup result, even if IP address is not configured on an interface. Route table lookup is enabled for MyMac packets even when SVI is not created.

Configuring a Static ARP Entry

You can configure a static ARP entry on the device to map IP addresses to MAC hardware addresses, including static multicast MAC addresses.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *number*
	Example:
	`switch(config)# interface ethernet 1/3`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**ip arp address** *ip-address mac-address* **Example:** `switch(config-if)# ip arp 192.168.1.1 0019.076c.1a78`	Associates an IP address with a MAC address as a static entry.

	Command or Action	Purpose
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Saves this configuration change.

Configuring Proxy ARP

Configure proxy ARP on the device to determine the media addresses of hosts on other networks or subnets.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *number*
	Example:
	`switch(config)# interface ethernet 1/3`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**ip proxy-arp** **Example:** `switch(config-if)# ip proxy-arp`	Enables proxy ARP on the interface.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Saves this configuration change.

Configuring Local Proxy ARP on Ethernet Interfaces

You can configure local proxy ARP on Ethernet interfaces.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

Step 2	**interface ethernet** *number*
	Example:
	`switch(config)# interface ethernet 1/3`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	[**no**]**ip local-proxy-arp** **Example:** `switch(config-if)# ip local-proxy-arp`	Enables Local Proxy ARP on the interface.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Saves this configuration change.

Configuring Local Proxy ARP on SVIs

You can configure local proxy ARP on SVIs.

Procedure

	Command or Action	Purpose

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

Step 2	**interface vlan vlan-id**
	Example:
	`switch(config)# interface vlan 5`
	`switch(config-if)#`
		Creates a VLAN interface and enters the configuration mode for the SVI.

Step 3	[**no**] **ip local-proxy-arp** **Example:** `switch(config-if)# ip local-proxy-arp`	Enables local proxy ARP on SVIs.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Gratuitous ARP

You can configure gratuitous ARP on an interface.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *number*
	Example:
	`switch(config)# interface ethernet 1/3`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**ip arp gratuitous** {**request** | **update**] **Example:** `switch(config-if)# ip arp gratuitous request`	Enables gratuitous ARP on the interface. Gratuitous ARP is enabled by default.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Saves this configuration change.

Configuring the Interface IP Address for the ICMP Source IP Field

You can configure an interface IP address for the ICMP source IP field to handle ICMP error messages.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ip source** {**ethernet** *slot/port* | **loopback** *number* | **port-channel** *number*} **icmp-errors** **Example:** `switch(config)# ip source loopback 0 icmp-errors`	Configures an interface IP address for the ICMP source IP field to route ICMP error messages.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves this configuration change.

Verifying the IPv4 Configuration

To display the IPv4 configuration information, perform one of the following tasks:

Command	Purpose
show ip adjacency	Displays the adjacency table.
show ip adjacency summary	Displays the summary of number of throttle adjacencies.
show ip arp	Displays the ARP table.
show ip arp summary	Displays the summary of the number of throttle adjacencies.
show ip interface	Displays IP-related interface information.
show ip arp statistics [**vrf** *default* | *management*]	Displays the ARP statistics.

CHAPTER 19

Configuring OSPFv2

This chapter describes how to configure Open Shortest Path First version 2 (OSPFv2) for IPv4 networks on the Cisco NX-OS device.

This chapter includes the following sections:

About OSPFv2

OSPFv2 is an IETF link-state protocol for IPv4 networks. An OSPFv2 router sends a special message, called a hello packet, out each OSPF-enabled interface to discover other OSPFv2 neighbor routers. Once a neighbor is discovered, the two routers compare information in the Hello packet to determine if the routers have compatible configurations. The neighbor routers try to establish adjacency, which means that the routers synchronize their link-state databases to ensure that they have identical OSPFv2 routing information. Adjacent routers share link-state advertisements (LSAs) that include information about the operational state of each link, the cost of the link, and any other neighbor information. The routers then flood these received LSAs out every OSPF-enabled interface so that all OSPFv2 routers eventually have identical link-state databases. When all OSPFv2 routers have identical link-state databases, the network is converged. Each router then uses Dijkstra’s Shortest Path First (SPF) algorithm to build its route table.

You can divide OSPFv2 networks into areas. Routers send most LSAs only within one area, which reduces the CPU and memory requirements for an OSPF-enabled router.

OSPFv2 supports IPv4.

Note OSPFv2 on Cisco NX-OS supports RFC 2328. This RFC introduced a different method to calculate route summary costs which is not compatible with the calculation used by RFC1583. RFC 2328 also introduced different selection criteria for AS-external paths. It is important_ to ensure that all routers support the same RFC. RFC. Use the **rfc1583compatibility** command if your network includes routers that are only compliant with RFC1583. The default supported RFC standard for OSPFv2 may be different for Cisco NX-OS and Cisco IOS. You must make adjustments to set the values identically. See the OSPF RFC Compatibility Mode Example section for more information.

Hello Packet

OSPFv2 routers periodically send Hello packets on every OSPF-enabled interface. The hello interval determines how frequently the router sends these Hello packets and is configured per interface. OSPFv2 uses Hello packets for the following tasks:

- Neighbor discovery
- Keepalives
- Bidirectional communications
- Designated router election (see the Designated Routers section)

The Hello packet contains information about the originating OSPFv2 interface and router, including the assigned OSPFv2 cost of the link, the hello interval, and optional capabilities of the originating router. An OSPFv2 interface that receives these Hello packets determines if the settings are compatible with the receiving interface settings. Compatible interfaces are considered neighbors and are added to the neighbor table (see the Neighbors section).

Hello packets also include a list of router IDs for the routers that the originating interface has communicated with. If the receiving interface sees its own router ID in this list, bidirectional communication has been established between the two interfaces.

OSPFv2 uses Hello packets as a keepalive message to determine if a neighbor is still communicating. If a router does not receive a Hello packet by the configured dead interval (usually a multiple of the hello interval), then the neighbor is removed from the local neighbor table.

Neighbors

An OSPFv2 interface must have a compatible configuration with a remote interface before the two can be considered neighbors. The two OSPFv2 interfaces must match the following criteria:

- Hello interval
- Dead interval
- Area ID (see the Areas, on page 226 section)
- Authentication
- Optional capabilities

If there is a match, the following information is entered into the neighbor table:

- Neighbor ID—The router ID of the neighbor.
- Priority—Priority of the neighbor. The priority is used for designated router election (see the Designated Routers, on page 225 section).
- State—Indication of whether the neighbor has just been heard from, is in the process of setting up bidirectional communications, is sharing the link-state information, or has achieved full adjacency.
- Dead time—Indication of the time since the last Hello packet was received from this neighbor.
- IP Address—The IP address of the neighbor.
- Designated Router—Indication of whether the neighbor has been declared as the designated router or as the backup designated router (see the Designated Routers, on page 225 section).
- Local interface—The local interface that received the Hello packet for this neighbor.

Adjacency

Not all neighbors establish adjacency. Depending on the network type and designated router establishment, some neighbors become fully adjacent and share LSAs with all their neighbors, while other neighbors do not. For more information, see the Designated Routers, on page 225 section.

Adjacency is established using Database Description (DD) packets, Link State Request (LSR) packets, and Link State Update (LSU) packets in OSPF. The Database Description packet includes just the LSA headers from the link-state database of the neighbor (see the Link-State Advertisements, on page 227 section). The local router compares these headers with its own link-state database and determines which LSAs are new or updated. The local router sends an LSR packet for each LSA that it needs new or updated information on. The neighbor responds with an LSU packet. This exchange continues until both routers have the same link-state information.

Designated Routers

Networks with multiple routers present a unique situation for OSPF. If every router floods the network with LSAs, the same link-state information is sent from multiple sources. Depending on the type of network, OSPFv2 might use a single router, the designated router (DR), to control the LSA floods and represent the network to the rest of the OSPFv2 area (see the Areas section). If the DR fails, OSPFv2 selects a backup designated router (BDR). If the DR fails, OSPFv2 uses the BDR.

Network types are as follows:

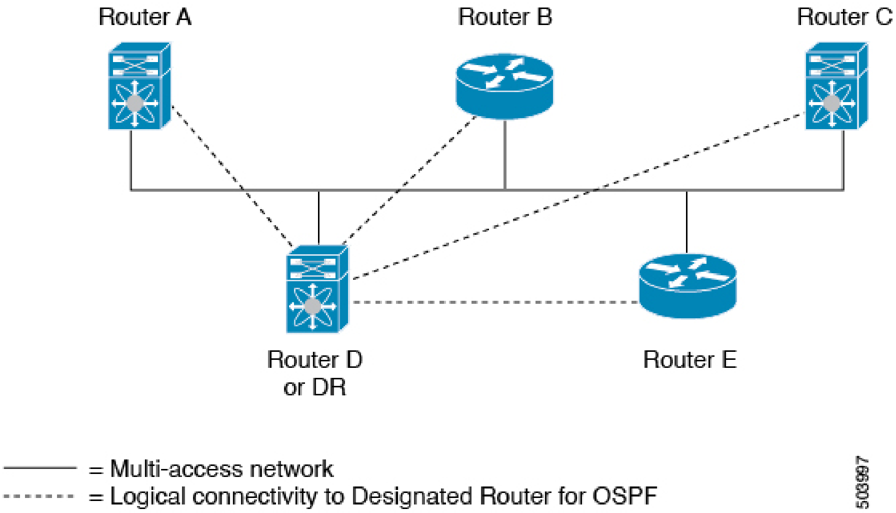
- Point-to-point—A network that exists only between two routers. All neighbors on a point-to-point network establish adjacency and there is no DR.
- Broadcast—A network with multiple routers that can communicate over a shared medium that allows broadcast traffic, such as Ethernet. OSPFv2 routers establish a DR and a BDR that controls LSA flooding on the network. OSPFv2 uses the well-known IPv4 multicast addresses 224.0.0.5 and a MAC address of 0100.5300.0005 to communicate with neighbors.

The DR and BDR are selected based on the information in the Hello packet. When an interface sends a Hello packet, it sets the priority field and the DR and BDR field if it knows who the DR and BDR are. The routers follow an election procedure based on which routers declare themselves in the DR and BDR fields and the priority field in the Hello packet. As a final tie breaker, OSPFv2 chooses the highest router IDs as the DR and BDR.

All other routers establish adjacency with the DR and the BDR and use the IPv4 multicast address 224.0.0.6 to send LSA updates to the DR and BDR. The figure below shows this adjacency relationship between all routers and the DR.

DRs are based on a router interface. A router might be the DR for one network and not for another network on a different interface.

Figure 13: DR in Multi-Access Network


Router A
Router B
Router C
Router D
or DR
Router E
—— = Multi-access network
------- = Logical connectivity to Designated Router for OSPF
503997

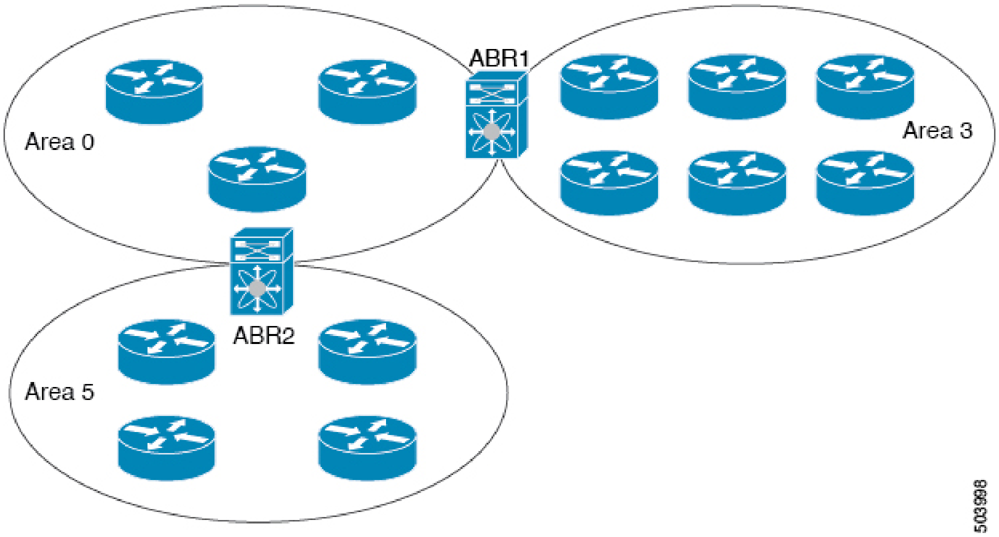
Areas

You can limit the CPU and memory requirements that OSPFv2 puts on the routers by dividing an OSPFv2 network into areas. An area is a logical division of routers and links within an OSPFv2 domain that creates separate subdomains. LSA flooding is contained within an area, and the link-state database is limited to links within the area. You can assign an area ID to the interfaces within the defined area. The Area ID is a 32-bit value that you can enter as a number or in dotted decimal notation, such as 10.2.3.1.

Cisco NX-OS always displays the area in dotted decimal notation.

If you define more than one area in an OSPFv2 network, you must also define the backbone area, which has the reserved area ID of 0. If you have more than one area, then one or more routers become area border routers (ABRs). The figure shows how an ABR connects to both the backbone area and at least one other defined area.

Figure 14: OSPFv2 Areas



The ABR has a separate link-state database for each area to which it connects. The ABR sends Network Summary (type 3) LSAs (see the Route Summarization section) from one connected area to the backbone area. The backbone area sends summarized information about one area to another area. In the OSPFv2 Areas Figure, Area 0 sends summarized information about Area 5 to Area 3.

OSPFv2 defines one other router type: the autonomous system boundary router (ASBR). This router connects an OSPFv2 area to another autonomous system. An autonomous system is a network controlled by a single technical administration entity. OSPFv2 can redistribute its routing information into another autonomous system or receive redistributed routes from another autonomous system. For more information, see the Advanced Features section.

Link-State Advertisements

OSPFv2 uses link-state advertisements (LSAs) to build its routing table.

Link-State Advertisement Types

OSPFv2 uses link-state advertisements (LSAs) to build its routing table.

The table shows the LSA types supported by Cisco NX-OS.

Table 16: Table 5-1 LSA Types

Type	Name	Description
1	Router LSA	LSA sent by every router. This LSA includes the state and the cost of all links and a list of all OSPFv2 neighbors on the link. Router LSAs trigger an SPF recalculation. Router LSAs are flooded to local OSPFv2 area.
2	Network LSA	LSA sent by the DR. This LSA lists all routers in the multi-access network. Network LSAs trigger an SPF recalculation. See the Designated Routers section.

Type	Name	Description
3	Network Summary LSA	LSA sent by the area border router to an external area for each destination in the local area. This LSA includes the link cost from the area border router to the local destination. See the Areas section.
4	ASBR Summary LSA	LSA sent by the area border router to an external area. This LSA advertises the link cost to the ASBR only. See the Areas section.
5	AS External LSA	LSA generated by the ASBR. This LSA includes the link cost to an external autonomous system destination. AS External LSAs are flooded throughout the autonomous system. See the Areas section.
7	NSSA External LSA	LSA generated by the ASBR within a not-so-stubby area (NSSA). This LSA includes the link cost to an external autonomous system destination. NSSA External LSAs are flooded only within the local NSSA. See the Areas section.
9-11	Opaque LSAs	LSA used to extend OSPF. See the Opaque LSAs section.

Link Cost

Each OSPFv2 interface is assigned a link cost. The cost is an arbitrary number. By default, Cisco NX-OS assigns a cost that is the configured reference bandwidth divided by the interface bandwidth. By default, the reference bandwidth is 40 Gb/s. The link cost is carried in the LSA updates for each link.

Flooding and LSA Group Pacing

When an OSPFv2 router receives an LSA, it forwards that LSA out every OSPF-enabled interface, flooding the OSPFv2 area with this information. This LSA flooding guarantees that all routers in the network have identical routing information. LSA flooding depends on the OSPFv2 area configuration (see the Areas section). The LSAs are flooded based on the link-state refresh time (every 30 minutes by default). Each LSA has its own link-state refresh time.

You can control the flooding rate of LSA updates in your network by using the LSA group pacing feature. LSA group pacing can reduce high CPU or buffer usage. This feature groups LSAs with similar link-state refresh times to allow OSPFv2 to pack multiple LSAs into an OSPFv2 Update message.

By default, LSAs with link-state refresh times within 10 seconds of each other are grouped together. You should lower this value for large link-state databases or raise it for smaller databases to optimize the OSPFv2 load on your network.

Link-State Database

Each router maintains a link-state database for the OSPFv2 network. This database contains all the collected LSAs, and includes information on all the routes through the network. OSPFv2 uses this information to calculate the bast path to each destination and populates the routing table with these best paths.

LSAs are removed from the link-state database if no LSA update has been received within a set interval, called the MaxAge. Routers flood a repeat of the LSA every 30 minutes to prevent accurate link-state information from being aged out. Cisco NX-OS supports the LSA grouping feature to prevent all LSAs from refreshing at the same time. For more information, see the Flooding and LSA Group Pacing section.

Opaque LSAs

Opaque LSAs allow you to extend OSPF functionality. Opaque LSAs consist of a standard LSA header followed by application-specific information. This information might be used by OSPFv2 or by other applications. OSPFv2 uses Opaque LSAs to support OSPFv2 Graceful Restart capability (see the High Availability and Graceful Restart section). Three Opaque LSA types are defined as follows:

- LSA type 9—Flooded to the local network.
- LSA type 10—Flooded to the local area.
- LSA type 11—Flooded to the local autonomous system.

OSPFv2 and the Unicast RIB

OSPFv2 runs the Dijkstra shortest path first algorithm on the link-state database. This algorithm selects the best path to each destination based on the sum of all the link costs for each link in the path. The resultant shortest path for each destination is then put in the OSPFv2 route table. When the OSPFv2 network is converged, this route table feeds into the unicast RIB. OSPFv2 communicates with the unicast RIB to do the following:

- Add or remove routes
- Handle route redistribution from other protocols
- Provide convergence updates to remove stale OSPFv2 routes and for stub router advertisements (see the OSPFv2 Stub Router Advertisements section)

OSPFv2 also runs a modified Dijkstra algorithm for fast recalculation for summary and external (type 3, 4, 5, and 7) LSA changes.

Authentication

You can configure authentication on OSPFv2 messages to prevent unauthorized or invalid routing updates in your network. Cisco NX-OS supports two authentication methods:

- Simple password authentication
- MD5 authentication digest

You can configure the OSPFv2 authentication for an OSPFv2 area or per interface.

Simple Password Authentication

Simple password authentication uses a simple clear-text password that is sent as part of the OSPFv2 message. The receiving OSPFv2 router must be configured with the same clear-text password to accept the OSPFv2 message as a valid route update. Because the password is in clear text, anyone who can watch traffic on the network can learn the password.

Cryptographic Authentication

Cryptographic authentication uses an encrypted password for OSPFv2 authentication. The transmitter computes a code using the packet to be transmitted and the key string, inserts the code and the key ID in the packet, and transmits the packet. The receiver validates the code in the packet by computing the code locally using the received packet and the key string (corresponding to the key ID in the packet) configured locally.

Both message digest 5 (MD5) and hash-based message authentication code secure hash algorithm (HMAC-SHA) cryptographic authentication are supported.

MD5 Authentication

You should use MD5 authentication to authenticate OSPFv2 messages. You configure a password that is shared at the local router and all remote OSPFv2 neighbors. For each OSPFv2 message, Cisco NX-OS creates an MD5 one-way message digest based on the message itself and the encrypted password. The interface sends this digest with the OSPFv2 message. The receiving OSPFv2 neighbor validates the digest using the same encrypted password. If the message has not changed, the digest calculation is identical and the OSPFv2 message is considered valid.

MD5 authentication includes a sequence number with each OSPFv2 message to ensure that no message is replayed in the network.

HMAC-SHA Authentication

OSPFv2 supports RFC 5709 to allow the use of HMAC-SHA algorithms, which offer more security than MD5. The HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384. and HMAC-SHA-512 algorithms are supported for OSPFv2 authentication.

Advanced Features

Cisco NX-OS supports advanced OSPFv3 features that enhance the usability and scalability of OSPFv2 in the network.

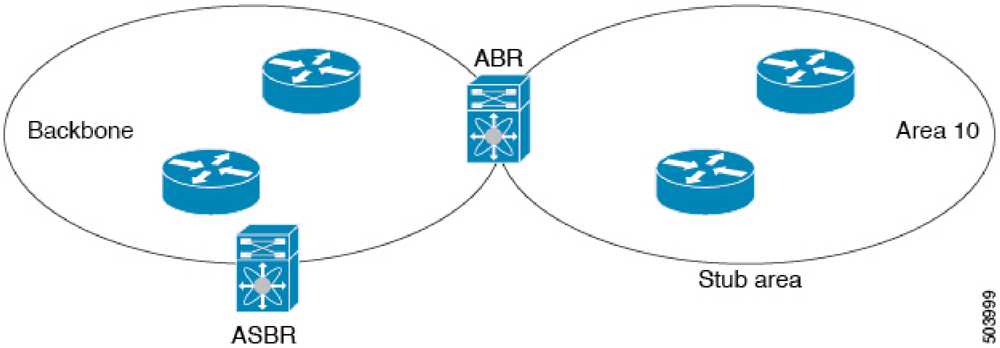
Stub Area

You can limit the amount of external routing information that floods an area by making it a stub area. A stub area is an area that does not allow AS External (type 5) LSAs (see the *Link State Advertisement* section). These LSAs are usually flooded throughout the local autonomous system to propagate external route information. Stub areas have the following requirements:

- All routers in the stub area are stub routers.
- No ASBR routers exist in the stub area.
- You cannot configure virtual links in the stub area.

The following figure shows an example of an OSPFv2 autonomous system where all routers in area 0.0.0.10 have to go through the ABR to reach external autonomous systems. Area 0.0.0.10 can be configured as a stub area.

Figure 15: Stub Area



Stub areas use a default route for all traffic that needs to go through the backbone area to the external autonomous system. The default route is 0.0.0.0 for IPv4.

Not-So-Stubby Area

A Not-so-Stubby Area (NSSA) is similar to a stub area, except that an NSSA allows you to import autonomous system external routes within an NSSA using redistribution. The NSSA ASBR redistributes these routes and generates NSSA External (type 7) LSAs that it floods throughout the NSSA. You can optionally configure the ABR that connects the NSSA to other areas to translate this NSSA External LSA to AS External (type 5) LSAs. The ABR then floods these AS External LSAs throughout the OSPFv2 autonomous system. Summarization and filtering are supported during the translation. See the Link-State Advertisements, on page 227 section for information about NSSA External LSAs.

You can, for example, use NSSA to simplify administration if you are connecting a central site using OSPFv2 to a remote site that is using a different routing protocol. Before NSSA, the connection between the corporate site border router and a remote router could not be run as an OSPFv2 stub area because routes for the remote site could not be redistributed into a stub area. With NSSA, you can extend OSPFv2 to cover the remote connection by defining the area between the corporate router and remote router as an NSSA.

The backbone Area 0 cannot be an NSSA.

Note OSPF is compliant with RFC 3101 section 2.5(3). When an Area Border Router attached to a Not-so-Stubby Area receives a default route LSA with P-bit clear, it should be ignored. OSPF had been previously adding the default route under these conditions.

If you have already designed your networks with RFC non-compliant behavior and expect a default route to be added on NSSA ABR, you will see a change in behavior when you upgrade.

If you decide to continue with the old behavior, you have the option to enable it with the **default-route nssa-abr pbit-clear** command.

Route Redistribution

OSPFv2 can learn routes from other routing protocols by using route redistribution. You configure OSPFv2 to assign a link cost for these redistributed routes or a default link cost for all redistributed routes.

Route redistribution uses route maps to control which external routes are redistributed. You must configure a route map with the redistribution to control which routes are passed into OSPFv2. A route map allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on. You can use route maps to modify parameters in the AS External (type 5) and NSSA External (type 7) LSAs before these external routes are advertised in the local OSPFv2 autonomous system. See the Configuring Route Policy Manager section, for information about configuring route maps.

Route Summarization

Because OSPFv2 shares all learned routes with every OSPF-enabled router, you might want to use route summarization to reduce the number of unique routes that are flooded to every OSPF-enabled router. Route summarization simplifies route tables by replacing more-specific addresses with an address that represents all the specific addresses. For example, you can replace 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one summary address, 10.1.0.0/16.

Typically, you would summarize at the boundaries of area border routers (ABRs). Although you could configure summarization between any two areas, it is better to summarize in the direction of the backbone so that the backbone receives all the aggregate addresses and injects them, already summarized, into other areas. The two types of summarization are as follows

- Inter-area route summarization
- External route summarization

You configure inter-area route summarization on ABRs, summarizing routes between areas in the autonomous system. To take advantage of summarization, you should assign network numbers in areas in a contiguous way to be able to lump these addresses into one range.

External route summarization is specific to external routes that are injected into OSPFv2 using route redistribution. You should make sure that external ranges that are being summarized are contiguous. Summarizing overlapping ranges from two different routers could cause packets to be sent to the wrong destination. Configure external route summarization on ASBRs that are redistributing routes into OSPF.

When you configure a summary address, Cisco NX-OS automatically configures a discard route for the summary address to prevent routing black holes and route loops.

High Availability and Graceful Restart

Cisco NX-OS provides a multilevel high-availability architecture. OSPFv2 supports stateful restart, which is also referred to as non-stop routing (NSR). If OSPFv2 experiences problems, it attempts to restart from its previous run-time state. The neighbors do not register any neighbor event in this case. If the first restart is not successful and another problem occurs, OSPFv2 attempts a graceful restart.

A graceful restart, or nonstop forwarding (NSF), allows OSPFv2 to remain in the data forwarding path through a process restart. When OSPFv2 needs to perform a graceful restart, it sends a link-local opaque (type 9) LSA, called a grace LSA (see the Opaque LSAs section). This restarting OSPFv2 platform is called NSF capable.

The grace LSA includes a grace period, which is a specified time that the neighbor OSPFv2 interfaces hold onto the LSAs from the restarting OSPFv2 interface. (Typically, OSPFv2 tears down the adjacency and discards all LSAs from a down or restarting OSPFv2 interface.) The participating neighbors, which are called NSF helpers, keep all LSAs that originate from the restarting OSPFv2 interface as if the interface was still adjacent.

When the restarting OSPFv2 interface is operational again, it rediscovers its neighbors, establishes adjacency, and starts sending its LSA updates again. At this point, the NSF helpers recognize that the graceful restart has finished.

Stateful restart is used in the following scenarios:

- First recovery attempt after the process experiences problems

Graceful restart is used in the following scenarios:

- Second recovery attempt after the process experiences problems within a 4-minute interval
- Manual restart of the process using the **restart ospf** command

OSPFv2 Stub Router Advertisements

You can configure an OSPFv2 interface to act as a stub router using the OSPFv2 Stub Router Advertisements feature. Use this feature when you want to limit the OSPFv2 traffic through this router, such as when you want to introduce a new router to the network in a controlled manner or limit the load on a router that is already overloaded. You might also want to use this feature for various administrative or traffic engineering reasons.

OSPFv2 stub router advertisements do not remove the OSPFv2 router from the network topology, but they do prevent other OSPFv2 routers from using this router to route traffic to other parts of the network. Only the traffic that is destined for this router or directly connected to this router is sent.

OSPFv2 stub router advertisements mark all stub links (directly connected to the local router) to the cost of the local OSPFv2 interface. All remote links are marked with the maximum cost (0xFFFF).

Multiple OSPFv2 Instances

Cisco Nexus® 3550-T switch supports multiple instances of the OSPFv2 protocol that run on the same node. You cannot configure multiple instances over the same interface. By default, every instance uses the same system router ID. You must manually configure the router ID for each instance if the instances are in the same OSPFv2 autonomous system. For the number of supported OSPFv2 instances, see the *Cisco Nexus® 3550-T Verified Scalability Guide*.

SPF Optimization

Cisco NX-OS optimizes the SPF algorithm in the following ways:

- Partial SPF for Network (type 2) LSAs, Network Summary (type 3) LSAs, and AS External (type 5) LSAs—When there is a change on any of these LSAs, Cisco NX-OS performs a faster partial calculation rather than running the whole SPF calculation.
- SPF timers—You can configure different timers for controlling SPF calculations. These timers include exponential backoff for subsequent SPF calculations. The exponential backoff limits the CPU load of multiple SPF calculations.

Prerequisites for OSPFv2

OSPFv2 has the following prerequisites:

- You must be familiar with routing fundamentals to configure OSPF.
- You are logged on to the switch.
- You have configured at least one interface for IPv4 that can communicate with a remote OSPFv2 neighbor.
- You have completed the OSPFv2 network strategy and planning for your network. For example, you must decide whether multiple areas are required.
- You have enabled the OSPF feature (see the Enabling OSPFv2 section).

Guidelines and Limitations for OSPFv2

OSPFv2 has the following configuration guidelines and limitations:

- If you enter the **no graceful-restart planned only** command, graceful restart is disabled.
- Cisco NX-OS displays areas in dotted decimal notation regardless of whether you enter the area in decimal or dotted decimal notation.
- All OSPFv2 routers must operate in the same RFC compatibility mode. OSPFv2 for Cisco Nexus® 3550-T switch complies with RFC 2328. Use the **rfc1583compatibility** command in router configuration mode if your network includes routers that support only RFC 1583.
- In scaled scenarios, when the number of interfaces and link-state advertisements in an OSPF process is large, the snmp-walk on OSPF MIB objects is expected to time out with a small-values timeout at the SNMP agent. If your observe a timeout on the querying SNMP agent while polling OSPF MIB objects, increase the timeout value on the polling SNMP agent.
- The following guidelines and limitations apply to the administrative distance feature:
    - When an OSPF route has two or more equal cost paths, configuring the administrative distance is non-deterministic for the **match ip route-source** command.
    - Configuring the administrative distance is supported only for the **match route-type**, **match ip address prefix-list**, and **match ip route-source prefix-list** commands. The other match statements are ignored.
    - There is no preference among the **match route-type**, **match ip address**, and **match ip route-source** commands for setting the administrative distance of OSPF routes. In this way, the behavior of the table map for setting the administrative distance in Cisco Nexus® 3550-T switch OSPF is different from that in Cisco IOS OSPF.
    - The discard route is always assigned an administrative distance of 220. No configuration in the table map applies to OSPF discard routes.
- The output of the **show run ospf** command might show the default values for some OSPF commands.
- Cisco Nexus® 3550-T switch does not forward OSPF neighbor discovery packets, OSPF neighbors are not discovered when Cisco Nexus® 3550-T is an intermediate switch.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Note Please note that ***Cisco Nexus 3550-T - 10.1(2t) release***, supports OSPFv2 only in the default VRF.

Default Settings for OSPFv2

The table lists the default settings for OSPFv2 parameters.

Table 17: Default OSPFv2 Parameters

Parameters	**Default**
Administrative distance	110
Hello interval	10 seconds
Dead interval	40 seconds
Discard routes	Enabled
Graceful restart grace period	60 seconds
OSPFv2 feature	Disabled
Stub router advertisement announce time	600 seconds
Reference bandwidth for link cost calculation	40 Gb/s
LSA minimal arrival time	1000 milliseconds
LSA group pacing	10 seconds
SPF calculation initial delay time	200 milliseconds
SPF minimum hold time	5000 milliseconds
SPF calculation initial delay time	1000 milliseconds

Configuring Basic OSPFv2

Configure OSPFv2 after you have designed your OSPFv2 network.

Enabling OSPFv2

You must enable the OSPFv2 feature before you can configure OSPFv2.

Procedure

	Command or Action	Purpose

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

Step 2	**feature ospf** **Example:** `switch(config)# feature ospf` **Example:**	Enables the OSPFv2 feature.
Step 3	(Optional) **show feature** **Example:** `switch(config)# show feature`	Displays enabled and disabled features.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

To disable the OSPFv2 feature and remove all associated configuration, use the no feature ospf command in global configuration mode:

Command	Purpose
no feature ospf **Example:** `switch(config)# no feature ospf`	Disables the OSPFv2 feature and removes all associated configuration.

Creating an OSPFv2 Instance

The first step in configuring OSPFv2 is to create an OSPFv2 instance. You assign a unique instance tag for this OSPFv2 instance. The instance tag can be any string.

For more information about OSPFv2 instance parameters, see the Configuring Advanced OSPFv2, on page 245 section.

Before you begin

Ensure that you have enabled the OSPF feature (see the Enabling OSPFv2 section).

Use the **show ip ospf** *instance-tag* command to verify that the instance tag is not in use.

OSPFv2 must be able to obtain a router identifier (for example, a configured loopback address) or you must configure the router ID option.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	[**no**]**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 201`
	`switch(config-router)`
		Creates a new OSPFv2 instance with the configured instance tag.

Step 3	(Optional) **router-id** *ip-address* **Example:** `switch(config-router)# router-id 192.0.2.1`	Configures the OSPFv2 router ID. This IP address identifies this OSPFv2 instance and must exist on a configured interface in the system.
Step 4	(Optional) **show ip ospf** *instance-tag* **Example:** `switch(config-router)# show ip ospf 201`	Displays OSPF information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

To remove the OSPFv2 instance and all associated configuration, use the no router ospf command in global configuration mode.

Command	Purpose
no router ospf *instance-tag* **Example:** `switch(config)# no router ospf 201`	Deletes the OSPF instance and the associated configuration.

Note This command does not remove the OSPF configuration in interface mode. You must manually remove any OSPFv2 commands configured in interface mode.

Configuring Optional Parameters on an OSPFv2 Instance

You can configure optional parameters for OSPF, see the Configuring Advanced OSPFv2, on page 245 section.

You can configure the following optional parameters for OSPFv2 in router configuration mode:

Before you begin

Ensure that you have enabled the OSPF feature, (see the Enabling OSPFv2 section).

OSPFv2 must be able to obtain a router identifier (for example, a configured loopback address) or you must configure the router ID option.

Procedure

	Command or Action	Purpose
Step 1	**distance** *number* **Example:** `switch(config-router)# distance 25`	Configures the administrative distance for this OSPFv2 instance. The range is from 1 to 255. The default is 110.
Step 2	**log-adjacency-changes [detail]** **Example:** `switch(config-router)# log-adjacency-changes`	Generates a system message whenever a neighbor changes state.
Step 3	**maximum-paths** *path-number* **Example:** `switch(config-router)# maximum-paths 4`	Configures the maximum number of equal OSPFv2 paths to a destination in the route table. This command is used for load balancing. The range is from 1 to 16. The default is 8.
Step 4	**distance** *number* **Example:** `switch(config-router)# distance 25`	Configures the administrative distance for this OSPFv2 instance. The range is from 1 to 255. The default is 110.
Step 5	**log-adjacency-changes** [**detail**] **Example:** `switch(config-router)# log-adjacency-changes`	Generates a system message whenever a neighbor changes state.
Step 6	**maximum-paths** *path-number* **Example:** `switch(config-router)# maximum-paths 4`	Configures the maximum number of equal OSPFv2 paths to a destination in the route table. This command is used for load balancing. The range is from 1 to 16. The default is 8. **Note** Cisco Nexus® 3550-T hardware installs only 1 path. ECMP is not supported in Cisco Nexus® 3550-T.

	Command or Action	Purpose
Step 7	**passive-interface default** **Example:** `switch(config-router)# passive-interface default`	Suppresses routing updates on all interfaces. This command is overridden by the VRF or interface command mode configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-router)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to create an OSPFv2 instance:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# copy running-config startup-config
```

Configuring Networks in OSPFv2

You can configure a network to OSPFv2 by associating it through the interface that the router uses to connect to that network (see the Neighbors section). You can add all networks to the default backbone area (Area 0), or you can create new areas using any decimal number or an IP address.

Note All areas must connect to the backbone area either directly or through a virtual link.

Note OSPF is not enabled on an interface until you configure a valid IP address for that interface.

Before you begin

Ensure that you have enabled the OSPF feature (see the Enabling OSPFv2 section).

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

Step 2	**interface** *interface-type slot/port*
	Example:
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**ip address** *ip-prefix/length* **Example:** `switch(config-if)# ip address 192.0.2.1/16`	Assigns an IP address and subnet mask to this interface.
Step 4	**ip router ospf** *instance-tag* **area** *area-id* [**secondaries none**] **Example:** `switch(config-if)# ip router ospf 201 area 0.0.0.15`	Adds the interface to the OSPFv2 instance and area.
Step 5	(Optional) **show ip ospf** *instance-tag* **interface** *interface-type slot/port* **Example:** `switch(config-if)# show ip ospf 201 interface ethernet 1/2`	Displays OSPF information.
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Saves this configuration change.
Step 7	(Optional) **ip ospf cost** *number* **Example:** `switch(config-if)# ip ospf cost 25`	Configures the OSPFv2 cost metric for this interface. The default is to calculate cost metric, based on reference bandwidth and interface bandwidth. The range is from 1 to 65535.
Step 8	(Optional) **ip ospf dead-interval** *seconds* **Example:** `switch(config-if)# ip ospf dead-interval 50`	Configures the OSPFv2 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.

Step 9	(Optional) **ip ospf hello-interval** *seconds*
	Example:
	`switch(config-if)# ip ospf`
	`hello-interval`
	`25`
		Configures the OSPFv2 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.

Step 10	(Optional) **[default | no] ip ospf passive-interface** **Example:**	Suppresses routing updates on the interface. This command overrides the router or VRF command mode configuration. The **default** option removes this interface mode command

	Command or Action	Purpose
	`switch(config-if)# ip ospf passive-interface`	and reverts to the router or VRF configuration, if present.
Step 11	(Optional) **ip ospf priority** *number* **Example:** `switch(config-if)# ip ospf priority 25`	Configures the OSPFv2 priority, used to determine the DR for an area. The range is from 0 to 255. The default is 1. See the Designated Routers section.
Step 12	(Optional) **ip ospf shutdown** **Example:** `switch(config-if)# ip ospf shutdown`	Shuts down the OSPFv2 instance on this interface.

Example

This example shows how to add a network area 0.0.0.10 in OSPFv2 instance 201:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router ospf 201 area 0.0.0.10
switch(config-if)# copy running-config startup-config
```

Use the **show ip ospf interface** command to verify the interface configuration. Use the **show ip ospf neighbor** command to see the neighbors for this interface.

Configuring Authentication for an Area

You can configure authentication for all networks in an area or for individual interfaces in the area. Interface authentication configuration overrides area authentication.

Before you begin

Ensure that you have enabled the OSPF feature , see the Enabling OSPFv2 section.

Ensure that all neighbors on an interface share the same authentication configuration, including the shared authentication key.

Create the key chain for this authentication configuration. See the *Cisco Nexus® 3550-T Security Configuration* section.

Note For OSPFv2, the key identifier in the **key** *key-id* command supports values from 2 to 255 only.

Procedure

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

	Command or Action	Purpose

	`switch# configure terminal`
	`switch(config)#`	

Step 2	**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		Creates a new OSPFv2 instance with the configured instance tag.

Step 3	**area** *area-id* **authentication** [**message-digest**] **Example:** `switch(config-router)# area 0.0.0.10 authentication`	Configures the authentication mode for an area.

Step 4	**interface** *interface-type slot/port*
	Example:
	`switch(config-router)# interface ethernet 1/2`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 5	(Optional) **ip ospf authentication-key** [**0** | **3**] *key* **Example:** `switch(config-if)# ip ospf authentication-key 0 mypass`	Configures simple password authentication for this interface. Use this command if the authentication is not set to key-chain or message-digest. 0 configures the password in clear text. 3 configures the password as 3DES encrypted.
Step 6	(Optional) **ip ospf message-digest-key** *key-id* **md5** [**0** | **3**] *key* **Example:** `switch(config-if)# ip ospf message-digest-key 21 md5 0 mypass`	Configures message digest authentication for this interface. Use this command if the authentication is set to message-digest. The key-id range is from 1 to 255. The MD5 option 0 configures the password in clear text and 3 configures the pass key as 3DES encrypted.
Step 7	(Optional) **show ip ospf** *instance-tag* **interface** *interface-type slot/port* **Example:** `switch(config-if)# show ip ospf 201 interface ethernet 1/2`	Displays OSPF information.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configuring Authentication for an Interface

You can configure authentication for all networks in an area or for individual interfaces in the area. Interface authentication configuration overrides area authentication.

Before you begin

Ensure that you have enabled the OSPF feature (see the Enabling OSPFv2, on page 236 section).

Ensure that all neighbors on an interface share the same authentication configuration, including the shared authentication key.

Create the key chain for this authentication configuration. See the *Cisco Nexus® 3550-T Security Configuration* section.

Note For OSPFv2, the key identifier in the **key** *key-id* command supports values from 2 to 255 only.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *interface-type slot/port*
	Example:
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**ip ospf authentication** [**message-digest**] **Example:** `switch(config-if)# ip ospf authentication`	Enables interface authentication mode for OSPFv2 for either cleartext or message-digest type. Overrides area-based authentication for this interface. All neighbors must share this authentication type.
Step 4	(Optional) **ip ospf authentication key-chain** *key-id* **Example:** `switch(config-if)# ip ospf authentication key-chain Test1`	Configures interface authentication to use key chains for OSPFv2. See the *Cisco Standalone Series NX-OS Security Configuration Guide*, for details on key chains.
Step 5	(Optional) **ip ospf authentication-key** [**0** | **3** | **7**] *key* **Example:** `switch(config-if)# ip ospf authentication-key 0 mypass`	Configures simple password authentication for this interface. Use this command if the authentication is not set to key-chain or message-digest. The options are as follows: • 0—Configures the password in clear text. • 3—Configures the pass key as 3DES encrypted. • 7—Configures the key as Cisco type 7 encrypted.

	Command or Action	Purpose
Step 6	(Optional) **ip ospf message-digest-key** *key-id* **md5** [**0** | **3** | **7**] *key* **Example:** `switch(config-if)# ip ospf message-digest-key 21 md5 0 mypass`	Configures message digest authentication for this interface. Use this command if the authentication is set to message-digest. The key-id range is from 1 to 255. The MD5 options are as follows: • 0—Configures the password in clear text. • 3—Configures the pass key as 3DES encrypted. • 7—Configures the key as Cisco type 7 encrypted.
Step 7	(Optional) **show ip ospf** *instance-tag* **interface** *interface-type slot/port* **Example:** `switch(config-if)# show ip ospf 201 interface ethernet 1/2`	Displays OSPF information.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to set an interface for simple, unencrypted passwords and set the password for Ethernet interface 1/2:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# ip router ospf 201 area 0.0.0.10
switch(config-if)# ip ospf authentication
switch(config-if)# ip ospf authentication-key 0 mypass
switch(config-if)# copy running-config startup-config
```

This example shows how to configure OSPFv2 HMAC-SHA-1 and MD5 cryptographic authentication:

```
switch# configure terminal
switch(config)# key chain chain1
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string 7 070724404206
switch(config-keychain-key)# accept-lifetime 01:01:01 Jan 01 2015 infinite
switch(config-keychain-key)# send-lifetime 01:01:01 Jan 01 2015 infinite
switch(config-keychain-key)# cryptographic-algorithm HMAC-SHA-1
switch(config-keychain-key)# exit
switch(config-keychain)# key 2
switch(config-keychain-key)# key-string 7 070e234f1f5b4a
switch(config-keychain-key)# accept-lifetime 10:51:01 Jul 24 2015 infinite
switch(config-keychain-key)# send-lifetime 10:51:01 Jul 24 2015 infinite
switch(config-keychain-key)# cryptographic-algorithm MD5
switch(config-keychain-key)# exit
switch(config-keychain)# exit

switch(config)# interface ethernet 1/1
switch(config-if)# ip router ospf 1 area 0.0.0.0
switch(config-if)# ip ospf authentication message-digest
switch(config-if)# ip ospf authentication key-chain chain1

switch(config-if)# show key chain chain1
Key-Chain chain1
Key 1 -- text 7 "070724404206"
cryptographic-algorithm HMAC-SHA-1
accept lifetime UTC (01:01:01 Jan 01 2015)-(always valid) [active]
send lifetime UTC (01:01:01 Jan 01 2015)-(always valid) [active]
Key 2 -- text 7 "070e234f1f5b4a"
cryptographic-algorithm MD
accept lifetime UTC (10:51:00 Jul 24 2015)-(always valid) [active]
send lifetime UTC (10:51:00 Jul 24 2015)-(always valid) [active]

switch(config-if)# show ip ospf interface ethernet 1/1
Ethernet1/1 is up, line protocol is up
IP address 11.11.11.1/24
Process ID 1 VRF default, area 0.0.0.3
Enabled by interface configuration
State BDR, Network type BROADCAST, cost 40
Index 6, Transmit delay 1 sec, Router Priority 1
Designated Router ID: 33.33.33.33, address: 11.11.11.3
Backup Designated Router ID: 1.1.1.1, address: 11.11.11.1
2 Neighbors, flooding to 2, adjacent with 2
Timer intervals: Hello 10, Dead 40, Wait 40, Retransmit 5
Hello timer due in 00:00:08
Message-digest authentication, using keychain key1 (ready)
Sending SA: Key id 2, Algorithm MD5
Number of opaque link LSAs: 0, checksum sum 0
```

Configuring Advanced OSPFv2

Configure OSPFv2 after you have designed your OSPFv2 network.

Configuring Filter Lists for Border Routers

You can separate your OSPFv2 domain into a series of areas that contain related networks. All areas must connect to the backbone area through an area border router (ABR). OSPFv2 domains can connect to external domains as well, through an autonomous system border router (ASBR).

ABRs have the following optional configuration parameters:

- Area range—Configures route summarization between areas. See the Configuring Route Summarization, on page 255 section.
- Filter list—Filters the Network Summary (type 3) LSAs that are allowed in from an external area.

ASBRs also support filter lists.

Before you begin

Ensure that you have enabled the OSPF feature. See the Enabling OSPFv2, on page 236 section).

Create the route map that the filter list uses to filter IP prefixes in incoming or outgoing Network Summary (type 3) LSAs. See the Configuring Route Policy Manager section, for more information. See the Areas, on page 226 section.

Procedure

	Command or Action	Purpose

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

Step 2	**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		Creates a new OSPFv2 instance with the configured instance tag.

Step 3	**area** *area-id* **filter-list route-map** *map-name* {**in** | **out**}
	Example:
	`switch(config-router)# area 0.0.0.10`
	`filter-list route-map FilterLSAs in`
		Filters incoming or outgoing Network Summary (type 3) LSAs on an ABR.

Step 4	(Optional) **show ip ospf policy statistics area** *id* **filter-list** {**in** | **out**}
	Example:
	`switch(config-router)# show ip ospf`
	`policy`
	`statistics area 0.0.0.10 filter-list in`
		Displays OSPF policy information.

Step 5	(Optional) **copy running-config startup-config**
	Example:
	`switch(config)# copy running-config`
	`startup-config`
		Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure a filter list in area 0.0.0.10:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 filter-list route-map FilterLSAs in
switch(config-router)# copy running-config startup-config
```

Configuring Stub Areas

You can configure a stub area for part of an OSPFv2 domain where external traffic is not necessary. Stub areas block AS External (type 5) LSAs and limit unnecessary routing to and from selected networks. See the Stub Area section. You can optionally block all summary routes from going into the stub area.

Before you begin

Ensure that you have enabled the OSPF feature. (see the Enabling OSPFv2 section).

Ensure that there are no virtual links or ASBRs in the proposed stub area.

Procedure

	Command or Action	Purpose

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

Step 2	**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		Creates a new OSPFv2 instance with the configured instance tag.

Step 3	**area** *area-id* **stub** **Example:** `switch(config-router)# area 0.0.0.10 stub`	Creates this area as a stub area.
Step 4	(Optional) **area** *area-id* **default-cost** *cost* **Example:** `switch(config-router)# area 0.0.0.10 default-cost 25`	Sets the cost metric for the default summary route sent into this stub area. The range is from 0 to 16777215. The default is 1.
Step 5	(Optional) **show ip ospf** *instance-tag* **Example:** `switch(config-router)# show ip ospf 201`	Displays OSPF information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a stub area:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 stub
switch(config-router)# copy running-config startup-config
```

Configuring a Totally Stubby Area

You can create a totally stubby area and prevent all summary route updates from going into the stub area.

To create a totally stubby area, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**area** *area-id* **stub no-summary** **Example:** `switch(config-router)# area 20 stub no-summary`	Creates this area as a totally stubby area.

Configuring NSSA

You can configure an NSSA for part of an OSPFv2 domain where limited external traffic is required. You can optionally translate this external traffic to an AS External (type 5) LSA and flood the OSPFv2 domain with this routing information. An NSSA can be configured with the following optional parameters:

- No redistribution—Redistributed routes bypass the NSSA and are redistributed to other areas in the OSPFv2 autonomous system. Use this option when the NSSA ASBR is also an ABR.
- Default information originate—Generates an NSSA External (type 7) LSA for a default route to the external autonomous system. Use this option on an NSSA ASBR if the ASBR contains the default route in the routing table. This option can be used on an NSSA ABR whether or not the ABR contains the default route in the routing table.
- Route map—Filters the external routes so that only those routes that you want are flooded throughout the NSSA and other areas.
- No summary—Blocks all summary routes from flooding the NSSA. Use this option on the NSSA ABR.
- Translate—Translates NSSA External LSAs to AS External LSAs for areas outside the NSSA. Use this command on an NSSA ABR to flood the redistributed routes throughout the OSPFv2 autonomous system. You can optionally suppress the forwarding address in these AS External LSAs. If you choose this option, the forwarding address is set to 0.0.0.0.

Note The translate option requires a separate **area** *area-id* **nssa**command, preceded by the **area** *area-id* **nssa** command that creates the NSSA and configures the other options.

Before you begin

Ensure that you have enabled the OSPF feature (see the Enabling OSPFv2, on page 236 section).

Ensure that there are no virtual links in the proposed NSSA and that it is not the backbone area.

Procedure

	Command or Action	Purpose

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

Step 2	**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		Creates a new OSPFv2 instance with the configured instance tag.

Step 3	**area** *area-id* **nssa [no-redistribution] [default-information-originate]originate** [**route-map** *map-name*]] [**no-summary**] **Example:** `switch(config-router)# area 0.0.0.10 nssa no-redistribution`	Creates this area as an NSSA.
Step 4	(Optional) **area** *area-id* **nssa translate type7** {**always | never**} [**suppress-fa**] **Example:** `switch(config-router)# area 0.0.0.10 nssa translate type7 always`	Configures the NSSA to translate AS External (type 7) LSAs to NSSA External (type 5) LSAs.
Step 5	(Optional) **area** *area-id* **default-cost** *cost* **Example:** `switch(config-router)# area 0.0.0.10 default-cost 25`	Sets the cost metric for the default summary route sent into this NSSA.
Step 6	(Optional) **show ip ospf** *instance-tag* **Example:** `switch(config-router)# show ip ospf 201`	Displays OSPF information.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create an NSSA that blocks all summary route updates:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 nssa no-summary
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that generates a default route:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 nssa default-info-originate
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that filters external routes and blocks all summary route updates:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 nssa route-map ExternalFilter no-summary
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA and then configure the NSSA to always translate AS External (type 7) LSAs to NSSA External (type 5) LSAs:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 nssa
switch(config-router)# area 0.0.0.10 nssa translate type 7 always
switch(config-router)# copy running-config startup-config
```

Configuring Multi-Area Adjacency

You can add more than one area to an existing OSPFv2 interface. The additional logical interfaces support multi-area adjacency.

Before you begin

You must enable OSPFv2 (see the Enabling OSPFv2, on page 236 section).

Ensure that you have configured a primary area for the interface (see the Configuring Networks in OSPFv2, on page 239 section).

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *interface-type slot/port* **Example:**	Enters interface configuration mode.

	Command or Action	Purpose

	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`	

Step 3	**ip router ospf** [*instance-tag*] **multi-area** *area-id* **Example:** `switch(config-if)# ip router ospf 201 multi-area 3`	Adds the interface to another area. **Note** The *instance-tag* argument is optional. If you do not specify an instance, the multi-area configuration is applied to the same instance that is configured for the primary area on that interface.
Step 4	(Optional) **show ip ospf** *instance-tag* **interface** *interface-type slot/port* **Example:** `switch(config-if)# show ip ospf 201 interface ethernet 1/2`	Displays OSPFv2 information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to add a second area to an OSPFv2 interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router ospf 201 area 0.0.0.10
switch(config-if)# ip router ospf 201 multi-area 20
switch(config-if)# copy running-config startup-config
```

Configuring Redistribution

You can redistribute routes that are learned from other routing protocols into an OSPFv2 autonomous system through the ASBR.

For redistributing the default route, you must specify the following parameter:

- Default information originate—Generates an autonomous system External (type 5) LSA for a default route to the external autonomous system.

Note Default information originate ignores **match** statements in the optional route map.

For non-default routes, you can configure the following optional parameters for route redistribution in OSPF:

- Default metric—Sets all redistributed routes to the same cost metric.

Note If you redistribute static routes, Cisco NX-OS requires the **default-information originate** command to successfully redistribute the default static route.

Before you begin

Enable the OSPF feature. See Enabling OSPFv2.

Create the necessary route maps used for redistribution.

Procedure

	Command or Action	Purpose

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

Step 2	**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		Creates a new OSPFv2 instance with the configured instance tag.

Step 3	**redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} **route-map** *map-name*
	Example:
	`switch(config-router)# redistribute bgp`
	`route-map FilterExternalBGP`
		Redistributes the selected protocol into OSPF through the configured route map.
		Note If you redistribute static routes, Cisco NX-OS also redistributes the default static route.

Step 4	**default-information originate** [**always**] [**route-map** *map-name*]
	Example:
	`switch(config-router)#`
	`default-information-originate route-map`
	`DefaultRouteFilter`
		Creates a default route into this OSPF domain if the default route exists in the RIB. Use the following optional keywords:
		• **always**—Always generate the default route of 0.0.0. even if the route does not exist in the RIB.
		• **route-map**—Generate the default route if the route map returns true.
		Note This command ignores **match** statements in the route map.

Step 5	**default-metric** [*cost*] **Example:**	Sets the cost metric for the redistributed routes. This command does not apply to directly

	Command or Action	Purpose
	`switch(config-router)# default-metric 25`	connected routes. Use a route map to set the default metric for directly connected routes.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to redistribute the Border Gateway Protocol (BGP) into OSPF:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# redistribute bgp route-map FilterExternalBGP
switch(config-router)# copy running-config startup-config
```

Limiting the Number of Redistributed Routes

Route redistribution can add many routes to the OSPFv2 route table. You can configure a maximum limit to the number of routes accepted from external protocols. OSPFv2 provides the following options to configure redistributed route limits:

- Fixed limit—Logs a message when OSPFv2 reaches the configured maximum. OSPFv2 does not accept any more redistributed routes. You can optionally configure a threshold percentage of the maximum where OSPFv2 logs a warning when that threshold is passed.
- Warning only—Logs a warning only when OSPFv2 reaches the maximum. OSPFv2 continues to accept redistributed routes.
- Withdraw—Starts the timeout period when OSPFv2 reaches the maximum. After the timeout period, OSPFv2 requests all redistributed routes if the current number of redistributed routes is less than the maximum limit. If the current number of redistributed routes is at the maximum limit, OSPFv2 withdraws all redistributed routes. You must clear this condition before OSPFv2 accepts more redistributed routes.
- You can optionally configure the timeout period.

Before you begin

Ensure that you have enabled the OSPF feature (see the Enabling OSPFv2, on page 236 section).

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

Step 2	**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		Creates a new OSPFv2 instance with the configured instance tag.

Step 3	**redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} **route-map** *map-name* **Example:** `switch(config-router)# redistribute bgp route-map FilterExternalBGP`	Redistributes the selected protocol into OSPF through the configured route map.
Step 4	**redistribute maximum-prefix** *max* [*threshold*] [**warning-only | withdraw** [*num-retries timeout*]] **Example:** `switch(config-router)# redistribute maximum-prefix 1000 75 warning-only`	Specifies a maximum number of prefixes that OSPFv2 distributes. The range is from 0 to 65536. Optionally specifies the following: • *threshold*—Percentage of maximum prefixes that trigger a warning message. • **warning-only**—Logs a warning message when the maximum number of prefixes is exceeded. • **withdraw**—Withdraws all redistributed routes. Optionally tries to retrieve the redistributed routes. The *num-retries* range is from 1 to 12. The *timeout* range is 60 to 600 seconds. The default is 300 seconds. Use the **clear ip ospf redistribution** command if all routes are withdrawn.
Step 5	(Optional) **show running-config ospf** **Example:** `switch(config-router)# show running-config ospf`	Displays the OSPFv2 configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to limit the number of redistributed routes into OSPF:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# redistribute bgp route-map FilterExternalBGP
switch(config-router)# redistribute maximum-prefix 1000 75
```

Configuring Route Summarization

You can configure route summarization for inter-area routes by configuring an address range that is summarized. You can also configure route summarization for external, redistributed routes by configuring a summary address for those routes on an ASBR. For more information, see the Route Summarization, on page 232 section.

Before you begin

Ensure that you have enabled the OSPF feature (see the Enabling OSPFv2, on page 236 section).

Procedure

	Command or Action	Purpose

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

Step 2	**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		Creates a new OSPFv2 instance with the configured instance tag.

Step 3	**area** *area-id* **range** *ip-prefix/length* [**no-advertise**] [**cost** *cost*] **Example:** `switch(config-router)# area 0.0.0.10 range 10.3.0.0/16`	Creates a summary address on an ABR for a range of addresses and optionally does not advertise this summary address in a Network Summary (type 3) LSA. The *cost* range is from 0 to 16777215.
Step 4	**summary-address** *ip-prefix/length* [**no-advertise** | **tag** *tag*] **Example:** `switch(config-router)# summary-address 10.5.0.0/16 tag 2`	Creates a summary address on an ASBR for a range of addresses and optionally assigns a tag for this summary address that can be used for redistribution with route maps.
Step 5	(Optional) **show ip ospf summary-address** **Example:** `switch(config-router)# show ip ospf summary-address`	Displays information about OSPF summary addresses.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create summary addresses between areas on an ABR:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)#  area 0.0.0.10 range 10.3.0.0/16
switch(config-router)# copy running-config startup-config
```

This example shows how to create summary addresses on an ASBR:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# summary-address 10.5.0.0/16
switch(config-router)# copy running-config startup-config
```

Configuring Stub Route Advertisements

Use stub route advertisements when you want to limit the OSPFv2 traffic through this router for a short time. For more information, see the OSPFv2 Stub Router Advertisements, on page 233 section.

Stub route advertisements can be configured with the following optional parameters:

- On startup—Sends stub route advertisements for the specified announce time.
- Wait for BGP—Sends stub router advertisements until BGP converges.

Note You should not save the running configuration of a router when it is configured for a graceful shutdown because the router continues to advertise a maximum metric after it is reloaded.

Before you begin

Ensure that you have enabled the OSPF feature (see the Enabling OSPFv2, on page 236 section).

Procedure

	Command or Action	Purpose

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

Step 2	**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		Creates a new OSPFv2 instance with the configured instance tag.

Step 3	**max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**include-stub**] [**on-startup** {*seconds* | **wait-for bgp** *tag*}] [**summary-lsa** [*max-metric-value*}] **Example:** `switch(config-router)# max-metric router-lsa`	Configures OSPFv2 stub route advertisements.

	Command or Action	Purpose
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable the stub router advertisements on startup for the default 600 seconds:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# max-metric router-lsa on-startup
switch(config-router)# copy running-config startup-config
```

Configuring the Administrative Distance of Routes

You can set the administrative distance of routes added by OSPFv2 into the RIB.

The administrative distance is a rating of the trustworthiness of a routing information source. A higher value indicates a lower trust rating. Typically, a route can be learned through more than one routing protocol. The administrative distance is used to discriminate between routes learned from more than one routing protocol. The route with the lowest administrative distance is installed in the IP routing table.

Before you begin

Ensure that you have enabled OSPF (see the Enabling OSPFv2 section).

See the guidelines and limitations for this feature in the Guidelines and Limitations for OSPFv2 section.

Procedure

	Command or Action	Purpose

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

Step 2	**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		Creates a new OSPFv2 instance with the configured instance tag.

Step 3	[**no**] **table-map** *map-name* **Example:** `switch(config-router)# table-map foo`	Configures the policy for filtering or modifying OSPFv2 routes before sending them to the RIB. You can enter up to 63 alphanumeric characters for the map name.

	Command or Action	Purpose

Step 4	**exit**
	Example:
	`switch(config-router)# exit`
	`switch(config)#`
		Exits router configuration mode.

Step 5	**route-map** *map-name* [**permit** | **deny**] [*seq*]
	Example:
	`switch(config)# route-map foo permit 10`
	`switch(config-route-map)#`
		Creates a route map or enters route-map configuration mode for an existing route map. Use *seq* to order the entries in a route map.
		Note The **permit** option enables you to set the distance. If you use the **deny** option, the default distance is applied.

Step 6	**match route-type** *route-type* **Example:** `switch(config-route-map)# match route-type external`	Matches against one of the following route types: • external—The external route (BGP, EIGRP, and OSPF type 1 or 2) • inter-area—OSPF inter-area route • internal—The internal route (including the OSPF intra- or inter-area) • intra-area—OSPF intra-area route • nssa-external—The NSSA external route (OSPF type 1 or 2) • type-1—The OSPF external type 1 route • type-2—The OSPF external type 2 route
Step 7	**match ip route-source prefix-list** *name* **Example:** `switch(config-route-map)# match ip route-source prefix-list p1`	Matches the IPv4 route source address or router ID of a route to one or more IP prefix lists. Use the **ip prefix-list** command to create the prefix list.
Step 8	**match ip address prefix-list** *name* **Example:** `switch(config-route-map)# match ip address prefix-list p1`	Matches against one or more IPv4 prefix lists. Use the **ip prefix-list** command to create the prefix list.
Step 9	**set distance** *value* **Example:** `switch(config-route-map)# set distance 150`	Sets the administrative distance of routes for OSPFv2. The range is from 1 to 255.
Step 10	(Optional) **copy running-config startup-config**	Saves this configuration change.

	Command or Action	Purpose
	Example: `switch(config-route-map)# copy running-config startup-config`	

Example

This example shows how to configure the OSPFv2 administrative distance for inter-area routes to 150, for external routes to 200, and for all prefixes in prefix list p1 to 190:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# table-map foo
switch(config-router)# exit
switch(config)# route-map foo permit 10
switch(config-route-map)# match route-type inter-area
switch(config-route-map)# set distance 150
switch(config-route-map)# exit
switch(config)# route-map foo permit 20
switch(config-route-map)# match route-type external
switch(config-route-map)# set distance 200
switch(config-route-map)# exit
switch(config)# route-map foo permit 30
switch(config-route-map)# match ip route-source prefix-list p1
switch(config-route-map)# match ip address prefix-list p1
switch(config-route-map)# set distance 190
```

Modifying the Default Timers

OSPFv2 includes a number of timers that control the behavior of protocol messages and shortest path first (SPF) calculations. OSPFv2 includes the following optional timer parameters:

- LSA arrival time—Sets the minimum interval allowed between LSAs that arrive from a neighbor. LSAs that arrive faster than this time are dropped.
- Pacing LSAs—Sets the interval at which LSAs are collected into a group and refreshed, checksummed, or aged. This timer controls how frequently LSA updates occur and optimizes how many are sent in an LSA update message (see the Flooding and LSA Group Pacing section).
- Throttle LSAs—Sets the rate limits for generating LSAs. This timer controls how frequently LSAs are generated after a topology change occurs.
- Throttle SPF calculation—Controls how frequently the SPF calculation is run.

At the interface level, you can also control the following timers:

- Retransmit interval—Sets the estimated time between successive LSAs
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.

See the Configuring Networks in OSPFv2 section for information about the hello interval and dead timer.

Before you begin

Ensure that you have enabled the OSPF feature (see the Enabling OSPFv2 section).

Procedure

	Command or Action	Purpose

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

Step 2	**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		Creates a new OSPFv2 instance with the configured instance tag.

Step 3	**timers lsa-arrival** *msec* **Example:** `switch(config-router)# timers lsa-arrival 2000`	Sets the LSA arrival time in milliseconds. The range is from 10 to 600000. The default is 1000 milliseconds.
Step 4	**timers lsa-group-pacing** *seconds* **Example:** `switch(config-router)# timers lsa-group-pacing 1800`	Sets the interval in seconds for grouping LSAs. The range is from 1 to 1800. The default is 240 seconds.
Step 5	**timers throttle lsa** *start-time hold-interval max-time* **Example:** `switch(config-router)# timers throttle lsa 3000 6000 6000`	Sets the rate limit in milliseconds for generating LSAs with the following timers: • *start-time*—The range is from 50 to 5000 milliseconds. The default value is 50 milliseconds. • *hold-interrval*—The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds. • *max-time*—The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds
Step 6	**timers throttle spf** *delay-time hold-time max-wait* **Example:** `switch(config-router)# timers throttle spf 3000 2000 4000`	Sets the SPF best path schedule initial delay time and the minimum hold time in seconds between SPF best path calculations. The range is from 1 to 600000. The default is no delay time and 5000 millisecond hold time.
Step 7	**interface** *type slot/port* **Example:**	Enters interface configuration mode.

	Command or Action	Purpose

	`switch(config)# interface ethernet 1/2`
	`switch(config-if)`	

Step 8	**ip ospf hello-interval** *seconds* **Example:** `switch(config-if)# ip ospf hello-interval 30`	Sets the hello interval for this interface. The range is from 1 to 65535. The default is 10.
Step 9	**ip ospf dead-interval** *seconds* **Example:** `switch(config-if)# ip ospf dead-interval 30`	Sets the dead interval for this interface. The range is from 1 to 65535.
Step 10	**ip ospf retransmit-interval** *seconds* **Example:** `switch(config-if)# ip ospf retransmit-interval 30`	Sets the estimated time in seconds between LSAs transmitted from this interface. The range is from 1 to 65535. The default is 5.

Step 11	**ip ospf transmit-delay** *seconds*
	Example:
	`switch(config-if)# ip ospf transmit-delay 450`
	`switch(config-if)#`
		Sets the estimated time in seconds to transmit an LSA to a neighbor. The range is from 1 to 450. The default is 1.

Step 12	(Optional) **show ip ospf** **Example:** `switch(config-if)# show ip ospf`	Displays information about OSPF.
Step 13	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to control LSA flooding with the lsa-group-pacing option:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# timers lsa-group-pacing 300
switch(config-router)# copy running-config startup-config
```

Configuring Graceful Restart

Graceful restart is enabled by default. You can configure the following optional parameters for graceful restart in an OSPFv2 instance:

- Grace period—Configures how long neighbors should wait after a graceful restart has started before tearing down adjacencies.
- Helper mode disabled—Disables helper mode on the local OSPFv2 instance. OSPFv2 does not participate in the graceful restart of a neighbor.
- Planned graceful restart only—Configures OSPFv2 to support graceful restart only in the event of a planned restart.

Before you begin

Ensure that you have enabled OSPF (see the Enabling OSPFv2 section).

Ensure that all neighbors are configured for graceful restart with matching optional parameters set.

Procedure

	Command or Action	Purpose

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

Step 2	**router ospf** *instance-tag*
	Example:
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		Creates a new OSPFv2 instance with the configured instance tag.

Step 3	**graceful-restart** **Example:** `switch(config-router)# graceful-restart`	Enables a graceful restart. A graceful restart is enabled by default.
Step 4	(Optional) **graceful-restart grace-period** *seconds* **Example:** `switch(config-router)# graceful-restart grace-period 120`	Sets the grace period, in seconds. The range is from 5 to 1800. The default is 60 seconds.
Step 5	(Optional) **graceful-restart helper-disable** **Example:** `switch(config-router)# graceful-restart helper-disable`	Disables helper mode. This feature is enabled by default.
Step 6	(Optional) **graceful-restart planned-only** **Example:** `switch(config-router)# graceful-restart planned-only`	Configures a graceful restart for planned restarts only.
Step 7	(Optional) **show ip ospf** *instance-tag* **Example:**	Displays OSPF information.

	Command or Action	Purpose
	`switch(config-router)# show ip ospf 201`	
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable a graceful restart if it has been disabled and set the grace period to 120 seconds:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# graceful-restart
switch(config-router)# graceful-restart grace-period 120
switch(config-router)# copy running-config startup-config
```

Restarting an OSPFv2 Instance

You can restart an OSPv2 instance. This action clears all neighbors for the instance.

To restart an OSPFv2 instance and remove all associated neighbors, use the following command:

Procedure

	Command or Action	Purpose
Step 1	**restart ospf** *instance-tag* **Example:** `switch(config)# restart ospf 201`	Restarts the OSPFv2 instance and removes all neighbors.

Verifying the OSPFv2 Configuration

To display the OSPFv2 configuration, perform one of the following tasks:

Command	Purpose
show ip ospf [*instance-tag*]	Displays information about one or more OSPF routing instances. The output includes the following area-level counts: • Interfaces in this area—A count of all interfaces added to this area (configured interfaces). • Active interfaces—A count of all interfaces considered to be in router link states and SPF (UP interfaces). • Passive interfaces—A count of all interfaces considered to be OSPF passive (no adjacencies will be formed). • Loopback interfaces—A count of all local loopback interfaces.
show ip ospf border-routers [**vrf** {**default** | **management** }]	Displays the OSPFv2 border router configuration.
show ip ospf database [**vrf** { **default** | **management**}]	Displays the OSPFv2 link-state database summary.
show ip ospf interface *number* [**vrf** {**default** | **management** }]	Displays OSPFv2-related interface information.
show ip ospf lsa-content-changed-list *neighbor-id interface - type number* [**vrf** {**default** | **management** }]	Displays the OSPFv2 LSAs that have changed.
show ip ospf neighbors [*neighbor-id*] [**detail**] [*interface - type number*] [**vrf** { **default** | **management** }] [**summary**]	Displays the list of OSPFv2 neighbors.
show ip ospf request-list *neighbor-id interface - type number* [**vrf** {**default** | **management** }]	Displays the list of OSPFv2 link-state requests.
show ip ospf retransmission-list *neighbor-id interface - type number* [**vrf** { **default** | **management** }]	Displays the list of OSPFv2 link-state retransmissions.
show ip ospf route [*ospf-route*] [**summary**] [**vrf** { **default** | **management** }]	Displays the internal OSPFv2 routes.
show ip ospf summary-address [**vrf** {**default** | **management** }]	Displays information about the OSPFv2 summary addresses.
show ip ospf vrf {**default** | **management** }	Displays information about the VRF-based OSPFv2 configuration.
show running-configuration ospf	Displays the current running OSPFv2 configuration.

Monitoring OSPFv2

To display OSPFv2 statistics, use the following commands:

Command	Purpose
show ip ospf policy statistics area *area-id* **filter list** {**in** | **out**} [**vrf** {**default** | **management**}]	Displays the OSPFv2 route policy statistics for an area.
show ip policy statistics redestribute {**bgp** *id* | **direct** | **ospf** *id* | **static**} [**vrf** {**default** | **management**}]	Displays the OSPFv2 route policy statistics.
show ip ospf statistics [**vrf** { **default** | **management**}]	Displays the OSPFv2 event counters.
show ip ospf traffic [*interface-type number*] [**vrf** {**default** | **management**}]	Displays the OSPFv2 packet counters.

Configuration Examples for OSPFv2

The following example shows how to configure OSPFv2:

```
feature ospf
router ospf 201
 router-id 290.0.2.1
interface ethernet 1/2
 ip router ospf 201 area 0.0.0.10
 ip ospf authentication
 ip ospf authentication-key 0 mypass
```

OSPF RFC Compatibility Mode Example

The following example shows how to configure OSPF to be compatible with routers that comply with RFC 1583:

Note You must configure RFC 1583 compatibility on any VRF that connects to routers running only RFC 1583 compatible OSPF.

```
switch# configure terminal
switch(config)# feature ospf
switch(config)# router ospf Test1
switch(config-router)# rfc1583compatibility
```

Additional References

For additional information related to implementing OSPF, see the following sections:

Related Documents for OSPFv2

Related Topic	**Document Title**
Keychains	*Cisco Nexus® 3550-T Security Configuration* section
Route maps	The Configuring Route Policy Manager section

MIBs

MIBs	**MIBs Link**
MIBs related to OSPFv2	To locate and download supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

CHAPTER 20

Configuring Basic BGP

This chapter describes how to configure Border Gateway Protocol (BGP) on the Cisco NX-OS device.

This chapter includes the following sections:

About Basic BGP

Cisco NX-OS supports BGP version 4, which includes multiprotocol extensions that allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families. BGP uses TCP as a reliable transport protocol to create TCP sessions with other BGP-enabled devices.

BGP uses a path-vector routing algorithm to exchange routing information between BGP-enabled networking devices or BGP speakers. Based on this information, each BGP speaker determines a path to reach a particular destination while detecting and avoiding paths with routing loops. The routing information includes the actual route prefix for a destination, the path of autonomous systems to the destination, and other path attributes.

BGP selects a single path, by default, as the best path to a destination host or network. Each path carries well-known mandatory, well-known discretionary, and optional transitive attributes that are used in BGP best-path analysis. You can influence BGP path selection by altering some of these attributes by configuring BGP policies. See the Route Policies and Resetting BGP Sessions, on page 285 section for more information.

BGP also supports load balancing. See the BGP Best-Path Selection, on page 289 section for more information.

Note Cisco Nexus 3550-T hardware does not supports installing ECMP routes.

BGP Autonomous Systems

An autonomous system (AS) is a network controlled by a single administration entity. An autonomous system forms a routing domain with one or more interior gateway protocols (IGPs) and a consistent set of routing policies. BGP supports 16-bit and 32-bit autonomous system numbers.

Separate BGP autonomous systems dynamically exchange routing information through external BGP (eBGP) peering sessions. BGP speakers within the same autonomous system can exchange routing information through internal BGP (iBGP) peering sessions.

4-Byte AS Number Support

BGP supports 2-byte autonomous system (AS) numbers in plain-text notation or as.dot notation and 4-byte AS numbers in plain-text notation.

Administrative Distance

An administrative distance is a rating of the trustworthiness of a routing information source. By default, BGP uses the administrative distances shown in the table.

Table 18: BGP Default Administrative Distances

Distance	Default Value	Function
External	20	Applied to routes learned from eBGP.
Internal	200	Applied to routes learned from iBGP.
Local	220	Applied to routes originated by the router.

Note The administrative distance does not influence the BGP path selection algorithm, but it does influence whether BGP-learned routes are installed in the IP routing table.

BGP Peers

A BGP speaker does not discover another BGP speaker automatically. You must configure the relationships between BGP speakers. A BGP peer is a BGP speaker that has an active TCP connection to another BGP speaker.

BGP Sessions

BGP uses TCP port 179 to create a TCP session with a peer. When a TCP connection is established between peers, each BGP peer initially exchanges all of its routes—the complete BGP routing table—with the other peer. After this initial exchange, the BGP peers send only incremental updates when a topology change occurs in the network or when a routing policy change occurs. In the periods of inactivity between these updates, peers exchange special messages called keepalives. The hold time is the maximum time limit that can elapse between receiving consecutive BGP update or keepalive messages.

Cisco NX-OS supports the following peer configuration options:

- Individual IPv4 address—BGP establishes a session with the BGP speaker that matches the remote address and AS number.
- IPv4 prefix peers for a single AS number—BGP establishes sessions with BGP speakers that match the prefix and the AS number.
- Dynamic AS number prefix peers—BGP establishes sessions with BGP speakers that match the prefix and an AS number from a list of configured AS numbers.

Dynamic AS Numbers for Prefix Peers and Interface Peers

Cisco NX-OS accepts a range or list of AS numbers to establish BGP sessions. For example, if you configure BGP to use IPv4 prefix 192.0.2.0/8 and AS numbers 33, 66, and 99, BGP establishes a session with 192.0.2.1 with AS number 66 but rejects a session from 192.0.2.2 with AS number 50.

Cisco NX-OS does not associate prefix peers with dynamic AS numbers as either interior BGP (iBGP) or external BGP (eBGP) sessions until after the session is established. See the "Configuring Advanced BGP" chapter for more information on iBGP and eBGP.

Note The dynamic AS number prefix peer configuration overrides the individual AS number configuration that is inherited from a BGP template. For more information, see the "Configuring Advanced BGP" chapter.

BGP Router Identifier

To establish BGP sessions between peers, BGP must have a router ID, which is sent to BGP peers in the OPEN message when a BGP session is established. The BGP router ID is a 32-bit value that is often represented by an IPv4 address. You can configure the router ID. By default, Cisco NX-OS sets the router ID to the IPv4 address of a loopback interface on the router. If no loopback interface is configured on the router, the software chooses the highest IPv4 address configured to a physical interface on the router to represent the BGP router ID. The BGP router ID must be unique to the BGP peers in a network.

If BGP does not have a router ID, it cannot establish any peering sessions with BGP peers.

BGP and the Unicast RIB

BGP communicates with the unicast routing information base (unicast RIB) to store IPv4 routes in the unicast routing table. After selecting the best path, if BGP determines that the best path change needs to be reflected in the routing table, it sends a route update to the unicast RIB.

BGP receives route notifications regarding changes to its routes in the unicast RIB. It also receives route notifications about other protocol routes to support redistribution.

BGP also receives notifications from the unicast RIB regarding next-hop changes. BGP uses these notifications to keep track of the reachability and IGP metric to the next-hop addresses.

Whenever the next-hop reachability or IGP metrics in the unicast RIB change, BGP triggers a best-path recalculation for affected routes.

Prerequisites for BGP

BGP has the following prerequisites:

- You must enable BGP (see the Enabling BGP section).
- You should have a valid router ID configured on the system.
- You must have an AS number, either assigned by a Regional Internet Registry (RIR) or locally administered.
- You must configure at least one IGP that is capable of recursive next-hop resolution.
- You must configure an address family under a neighbor for the BGP session establishment.

Guidelines and Limitations for Basic BGP

BGP has the following configuration guidelines and limitations:

- With sufficient scale (such as - hundreds of peers and thousands of routes per peer) the Graceful Restart mechanism may fail because the default 5 minute stale-path timer might not be enough for BGP convergence to complete before the timer expires. Use the following command to verify the actual time taken for the convergence process:

```
switch# show bgp vrf all all neighbors | in First|RIB
  Last End-of-RIB received 0.022810 after session start
  Last End-of-RIB sent 00:08:36 after session start
  First convergence 00:08:36 after session start with 398002 routes sent
```

Note In Cisco Nexus 3550-T BGP is supported only in default VRF.

- The dynamic AS number prefix peer configuration overrides the individual AS number configuration that is inherited from a BGP template.
- If you configure a dynamic AS number for prefix peers in an AS confederation, BGP establishes sessions with only the AS numbers in the local confederation.
- BGP sessions that are created through a dynamic AS number prefix peer ignore any configured eBGP multihop time-to-live (TTL) value or a disabled check for directly connected peers.
- Configure a router ID for BGP to avoid automatic router ID changes and session flaps.
- Use the maximum-prefix configuration option per peer to restrict the number of routes that are received and system resources used.
- Configure the update source to establish a session with BGP/eBGP multihop sessions.
- Specify a BGP policy if you configure redistribution.
- If you decrease the keepalive and hold timer values, you might experience BGP session flaps.
- Although the **show ip bgp** commands are available for verifying the BGP configuration, Cisco recommends that you use the **show bgp** commands instead.

- BGP prefix independent convergence (PIC) edge feature is not supported in Cisco Nexus 3550-T.

Default Settings

Table 19: Default BGP Parameters

Parameters	Default
BGP feature	Disabled
Keep alive interval	60 seconds
Hold timer	180 seconds
Auto-summary	Always disabled
Synchronization	Always disabled

CLI Configuration Modes

The following sections describe how to enter each of the CLI configuration modes for BGP. From a mode, you can enter the **?** command to display the commands available in that mode.

Global Configuration Mode

Use global configuration mode to create a BGP process and configure advanced features such as AS confederation and route dampening.

This example shows how to enter router configuration mode:

```
switch# configuration
switch(config)# router bgp 64496
switch(config-router)#
```

Neighbor Configuration Mode

Cisco NX-OS provides the neighbor configuration mode to configure BGP peers. You can use neighbor configuration mode to configure all parameters for a peer.

The following example shows how to enter neighbor configuration mode:

```
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.1
switch(config-router-neighbor)#
```

Configuring Basic BGP

To configure a basic BGP, you must enable BGP and configure a BGP peer. Configuring a basic BGP network consists of a few required tasks and many optional tasks. You must configure a BGP routing process and BGP peers.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Enabling BGP

You must enable BGP before you can configure BGP.

Procedure

	Command or Action	Purpose

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

Step 2	**[no] feature bgp** **Example:** `switch(config)# feature bgp`	Enables BGP. Use the **no** form of this command to disable this feature.
Step 3	(Optional) **show feature** **Example:** `switch(config)# show feature`	Displays enabled and disabled features.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves this configuration change.

Creating a BGP Instance

You can create a BGP instance and assign a router ID to the BGP instance. For more information, see the BGP Router Identifier section.

Before you begin

- You must enable BGP (see the Enabling BGP section).
- BGP must be able to obtain a router ID (for example, a configured loopback address).

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
		Use the **no** option with this command to remove the BGP process and the associated configuration.

Step 3	(Optional) **router-id** *ip-address*
	Example:
	`switch(config-router)# router-id`
	`192.0.2.255`
		Configures the BGP router ID. This IP address identifies this BGP speaker.

Step 4	(Optional) **address-family** {**ipv4**} {**unicast**}
	Example:
	`switch(config-router)# address-family`
	`ipv4 unicast`
	`switch(config-router-af)#`
		Enters global address family configuration mode for the IPv4 address family.
		Note In Cisco Nexus 3550-T BGP supports only IPv4 Unicast address family.

Step 5	(Optional) **network** {*ip-address/length* | *ip-address* **mask** *mask*} [**route-map** *map-name*]
	Example:
	`switch(config-router-af)# network`
	`10.10.10.0/24`
	Example:
	`switch(config-router-af)# network`
	`10.10.10.0 mask 255.255.255.0`
		Specifies a network as local to this autonomous system and adds it to the BGP routing table.
		For exterior protocols, the network command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.

Step 6	(Optional) **show bgp all** **Example:** `switch(config-router-af)# show bgp all`	Displays information about all BGP address families.

Step 7	(Optional) **copy running-config startup-config**
	Example:
	`switch(config-router-af)# copy`
	`running-config`
	`startup-config`
		Saves this configuration change.

Example

This example shows how to enable BGP with the IPv4 unicast address family and manually add one network to advertise:

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# network 192.0.2.0
switch(config-router-af)# copy running-config startup-config
```

Restarting a BGP Instance

You can restart a BGP instance and clear all peer sessions for the instance.

To restart a BGP instance and remove all associated peers, use the following command:

Procedure

	Command or Action	Purpose
Step 1	**restart bgp***instance-tag* **Example:** `switch(config)# restart bgp 201`	Restarts the BGP instance and resets or reestablishes all peering sessions.

Shutting Down BGP

You can shut down the BGP protocol and gracefully disable BGP while retaining the configuration.

To shut down BGP, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**shutdown** **Example:** `switch(config-router)# shutdown`	Restarts the BGP instance and resets or reestablishes all peering sessions.

Configuring BGP Peers

You can configure a BGP peer within a BGP process. Each BGP peer has an associated keepalive timer and hold timers. You can set these timers either globally or for each BGP peer. A peer configuration overrides a global configuration.

Note You must configure the address family under neighbor configuration mode for each peer.

Before you begin

- You must enable BGP (see the Enabling BGP section).

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 3	**neighbor** {*ip-address*} **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor`
	`209.165.201.1 remote-as 64497`
	`switch(config-router-neighbor)#`
		Configures the IPv4 address and AS number for a remote BGP peer. *The ip-address* format is x.x.x.x. The format is A:B::C:D.

Step 4	**neighbor-as** *as-number*
	Example:
	`switch(config-router-neighbor)#`
	`remote-as 64497`
		Configures the AS number for a remote BGP peer.

Step 5	(Optional) **description***text*
	Example:
	`switch(config-router-neighbor)#`
	`description Peer Router B`
	`switch(config-router-neighbor)#`
		Adds a description for the neighbor. The description is an alphanumeric string up to 80 characters.

Step 6	(Optional) **timers***keepalive-time hold-time*
	Example:
	`switch(config-router-neighbor)# timers`
	`30 90`
		Adds the keepalive and hold time BGP timer values for the neighbor. The range is from 0 to 3600 seconds. The default is 60 seconds for the keepalive time and 180 seconds for the hold time.

Step 7	(Optional) **shutdown** **Example:** `switch(config-router-neighbor)# shutdown`	Administratively shuts down this BGP neighbor. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

Step 8	**address-family**{**ipv4**} {**unicast**}
	Example:
	`switch(config-router-neighbor)#`
	`address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Enters neighbor address family configuration mode for the unicast IPv4 address family.

	Command or Action	Purpose
Step 9	(Optional) **weight** *value* **Example:** `switch(config-router-neighbor-af)# weight 100`	Sets the default weight for routes from this neighbor. The range is from 0 to 65535. All routes learned from this neighbor have the assigned weight initially. The route with the highest weight is chosen as the preferred route when multiple routes are available to a particular network. The weights assigned with the **set weight route-map** command override the weights assigned with this command. If you specify a BGP peer policy template, all the members of the template inherit the characteristics configured with this command.
Step 10	(Optional) **show bgp** {**ipv4**} {**unicast**} **neighbors** **Example:** `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	Displays information about BGP peers.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	Saves this configuration change.

Example

The following example shows how to configure a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.1 remote-as 64497
switch(config-router-neighbor)# description Peer Router B
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor)# weight 100
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring Dynamic AS Numbers for Prefix Peers

You can configure multiple BGP peers within a BGP process. You can limit BGP session establishment to a single AS number or multiple AS numbers in a route map.

BGP sessions configured through dynamic AS numbers for prefix peers ignore the **ebgp-multihop** command and the **disable-connected-check** command.

You can change the list of AS numbers in the route map, but you must use the no neighbor command to change the route-map name. Changes to the AS numbers in the configured route map affect only new sessions.

Before you begin

- You must enable BGP (see the Enabling BGP section).

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 3	**neighbor** *prefix* **remote-as route-map** *map-name*
	Example:
	`switch(config-router)# neighbor 192.0.2.0/8 remote-as routemap BGPPeers`
	`switch(config-router-neighbor)#`
		Configures the IPv4 prefix and a route map for the list of accepted AS numbers for the remote BGP peers. The *prefix* format for IPv4 is x.x.x.x/length. The length range is from 1 to 32.
		The *map-name* can be any case-sensitive, alphanumeric string up to 63 characters.

Step 4	**neighbor-as** *as-number* **Example:** `switch(config-router-neighbor)# remote-as 64497`	Configures the AS number for a remote BGP peer.
Step 5	(Optional) **show bgp** {**ipv4** {**unicast** } **neighbors** **Example:** `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	Displays information about BGP peers.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to configure dynamic AS numbers for a prefix peer:

```
switch# configure terminal
switch(config)# route-map BGPPeers
switch(config-route-map)# match as-number 64496, 64501-64510
switch(config-route-map)# match as-number as-path-list List1, List2
switch(config-route-map)# exit
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.0/8 remote-as route-map BGPPeers
switch(config-router-neighbor)# description Peer Router B
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

Clearing BGP Information

To clear BGP information, use the following commands:

Command	Purpose
clear bgp all {*neighbor* | * | *as-number* | **peer-template** *name* | *prefix*}	Clears one or more neighbors from all address families. * clears all neighbors in all address families. The arguments are as follows: • *neighbor*—IPv4 address of a neighbor. • *as-number*— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. • *name*—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • *prefix*—IPv4 prefix. All neighbors within that prefix are cleared.
clear bgp all dampening	Clears route flap dampening networks in all address families.
clear bgp all flap-statistics	Clears route flap statistics in all address families.
clear bgp {**ipv4** } {**unicast**} **dampening**	Clears route flap dampening networks in the selected address family.
clear bgp {**ipv4** } {**unicast**} **flap-statistics**	Clears route flap statistics in the selected address family.

Command	Purpose
clear bgp {**ipv4** } {*neighbor* |* | *as-number* | **peer-template** *name* | *prefix*}	Clears one or more neighbors from the selected address family. * clears all neighbors in the address family. The arguments are as follows: • *neighbor*—IPv4 address of a neighbor. • *as-number*— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. • *name*—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • *prefix*—IPv4 prefix. All neighbors within that prefix are cleared.
clear bgp {**ip** {**unicast**}} {*neighbor* |* |*as-number* | **peer-template** *name* | *prefix*}	Clears one or more neighbors. * clears all neighbors in the address family. The arguments are as follows: • *neighbor*—IPv4 address of a neighbor. • *as-number*— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. • *name*—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • *prefix*—IPv4 prefix. All neighbors within that prefix are cleared.
clear bgp dampening [*ip-neighbor* | *ip-prefix*]	Clears route flap dampening in one or more networks. The arguments are as follows: • *ip-neighbor*—IPv4 address of a neighbor. • *ip-prefix*—IPv4. All neighbors within that prefix are cleared.
clear bgp flap-statistics [*ip-neighbor* | *ip-prefix*]	Clears route flap statistics in one or more networks. The arguments are as follows: • *ip-neighbor*—IPv4 address of a neighbor. • *ip-prefix*—IPv4. All neighbors within that prefix are cleared.

Verifying the Basic BGP Configuration

To display the BGP configuration, perform one of the following tasks:

Command	Purpose
show bgp all [summary]	Displays the BGP information for all address families.
show bgp convergence	Displays the BGP information for all address families.
show bgp {**ipv4** } {**unicast**} [*ip-address* **community** [**regexp** *expression* | [**community**] [**no-advertise**] [**no-export**] [**no-export-subconfed**]}	Displays the BGP routes that match a BGP community.
show bgp {**ipv4** } {**unicast**} [*ip-address*] **community-list** *list-name*	Displays the BGP routes that match a BGP community list.
show bgp {**ipv4** } {**unicast**} [*ip-address* | **extcommunity** [**regexp** *expression* | [**generic** [**non-transitive** | **transitive**] *aa4:nn* [**exact-match**]}	Displays the BGP routes that match a BGP extended community.
show bgp {**ipv4** } {**unicast**} [*ip-address* | **extcommunity-list** *list-name* [**exact-match**]}	Displays the BGP routes that match a BGP extended community list.
show bgp {**ipv4** } {**unicast**} [*ip-address* | {**dampening dampened-paths** [**regexp** *expression*]}	Displays the information for BGP route dampening. Use the **clear bgp dampening** command to clear the route flap dampening information.
show bgp {**ipv4** } {**unicast**} [*ip-address* | **history-paths** [**regexp** *expression*]	Displays the BGP route history paths.
show bgp {**ipv4** } {**unicast**} [*ip-address* | **filter-list** *list-name*	Displays the information for the BGP filter list.
show bgp {**ipv4** } {**unicast**} [*ip-address*] **neighbors** [*ip-address*]	Displays the information for BGP peers. Use the **clear bgp neighbors** command to clear these neighbors.
show bgp {**ipv4** } {**unicast**} [*ip-address*] **neighbors** [*ip-address*] {**nexthop** | **nexthop-database**}	Displays the information for the BGP route next hop.
show bgp paths	Displays the BGP path information.
show bgp {**ipv4** } {**unicast**} [*ip-address*] **policy** *name*	Displays the BGP policy information. Use the **clear bgp polic**y command to clear the policy information.
show bgp {**ipv4** } {**unicast**} [*ip-address*] **prefix-list** *list-name*	Displays the BGP routes that match the prefix list.
show bgp {**ipv4**} {**unicast**} [*ip-address*] **received-paths**	Displays the BGP paths stored for soft reconfiguration.
show bgp {**ipv4**} {**unicast**} [*ip-address*] **regexp** *expression*	Displays the BGP routes that match the AS_path regular expression.

Command	Purpose
show bgp {**ipv4**} {**unicast**} [*ip-address*] **route-map** *map-name*	Displays the BGP routes that match the route map.
show bgp peer-policy *name*	Displays the information about BGP peer policies.
show bgp peer-session *name* show bgp peer-session	Displays the information about BGP peer sessions.
show bgp peer-template *name*	Displays the information about BGP peer templates. Use the **clear bgp peer-template** command to clear all neighbors in a peer template.
show bgp process	Displays the BGP process information.
show {**ipv4**} **bgp** [*options*]	Displays the BGP status and configuration information.
show {**ipv4**} **mbgp** [*options*]	Displays the BGP status and configuration information.
show running-configuration bgp	Displays the current running BGP configuration.

Monitoring BGP Statistics

To display BGP statistics, use the following commands:

Command	Purpose
show bgp {**ipv4** } {**unicast**} [*ip-address*] **flap-statistics**	Displays the BGP route flap statistics. Use the **clear bgp flap-statistics command** to clear these statistics.
show bgp sessions	Displays the BGP sessions for all peers. Use the **clear bgp sessions** command to clear these statistics.
show bgp statistics	Displays the BGP statistics.

Configuration Examples for Basic BGP

This example shows a basic BGP configuration:

```
switch(config)# feature bgp
switch(config)# router bgp 64496
switch(config-router)# neighbor 10.10.10.10 remote-as 64496
switch(config-router-af)# next-hop-self
```

Related Topics

The following topics relate to BGP:

- Configuring Advanced BGP, on page 283

Where to Go Next

See Configuring Advanced BGP, on page 283, for details on the following features:

- Peer templates
- Route redistribution
- Route maps

CHAPTER 21

Configuring Advanced BGP

This chapter contains the following sections:

About Advanced BGP

BGP is an interdomain routing protocol that provides loop-free routing between organizations or autonomous systems. Cisco NX-OS supports BGP version 4. BGP version 4 includes multiprotocol extensions that allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families. BGP uses TCP as a reliable transport protocol to create TCP sessions with other BGP-enabled devices called BGP peers. When connecting to an external organization, the router creates external BGP (eBGP) peering sessions. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions.

Peer Templates

BGP peer templates allow you to create blocks of common configuration that you can reuse across similar BGP peers. Each block allows you to define a set of attributes that a peer then inherits. You can choose to override some of the inherited attributes as well, making it a very flexible scheme for simplifying the repetitive nature of BGP configurations.

Cisco NX-OS implements three types of peer templates:

- The peer-session template defines BGP peer session attributes, such as the transport details, remote autonomous system number of the peer, and session timers. A peer-session template can also inherit attributes from another peer-session template (with locally defined attributes that override the attributes from an inherited peer-session).

- A peer-policy template defines the address-family dependent policy aspects for a peer including the inbound and outbound policy, filter-lists, and prefix-lists. A peer-policy template can inherit from a set of peer-policy templates. Cisco NX-OS evaluates these peer-policy templates in the order specified by the preference value in the inherit configuration. The lowest number is preferred over higher numbers.
- The peer template can inherit the peer-session and peer-policy templates to allow for simplified peer definitions. It is not mandatory to use a peer template but it can simplify the BGP configuration by providing reusable blocks of configuration.

Authentication

You can configure authentication for a BGP neighbor session. This authentication method adds an MD5 authentication digest to each TCP segment sent to the neighbor to protect BGP against unauthorized messages and TCP security attacks.

Note The MD5 password must be identical between BGP peers.

Route Policies and Resetting BGP Sessions

You can associate a route policy to a BGP peer. Route policies use route maps to control or modify the routes that BGP recognizes. You can configure a route policy for inbound or outbound route updates. The route policies can match on different criteria, such as a prefix or AS_path attribute, and selectively accept or deny the routes. Route policies can also modify the path attributes.

When you change a route policy applied to a BGP peer, you must reset the BGP sessions for that peer. Cisco NX-OS supports the following three mechanisms to reset BGP peering sessions:

- Hard reset—A hard reset tears down the specified peering sessions, including the TCP connection, and deletes routes coming from the specified peer. This option interrupts packet flow through the BGP network. Hard reset is disabled by default.
- Soft reconfiguration inbound—A soft reconfiguration inbound triggers routing updates for the specified peer without resetting the session. You can use this option if you change an inbound route policy. Soft reconfiguration inbound saves a copy of all routes received from the peer before processing the routes through the inbound route policy. If you change the inbound route policy, Cisco NX-OS passes these stored routes through the modified inbound route policy to update the route table without tearing down existing peering sessions. Soft reconfiguration inbound can use significant memory resources to store the unfiltered BGP routes. Soft reconfiguration inbound is disabled by default.
- Route Refresh—A route refresh updates the inbound routing tables dynamically by sending route refresh requests to supporting peers when you change an inbound route policy. The remote BGP peer responds with a new copy of its routes that the local BGP speaker processes with the modified route policy. Cisco NX-OS automatically sends an outbound route refresh of prefixes to the peer.
- BGP peers advertise the route refresh capability as part of the BGP capability negotiation when establishing the BGP peer session. Route refresh is the preferred option and enabled by default.

Note BGP also uses route maps for route redistribution, route aggregation, route dampening, and other features.

eBGP

External BGP (eBGP) allows you to connect BGP peers from different autonomous systems to exchange routing updates. Connecting to external networks enables traffic from your network to be forwarded to other networks and across the Internet.

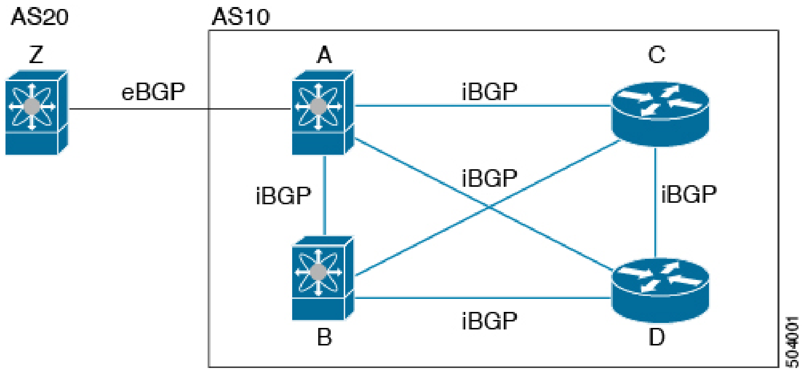
Typically eBGP peerings need to be over directly connected interfaces so that convergence will be faster when the interface goes down.

iBGP

Internal BGP (iBGP) allows you to connect BGP peers within the same autonomous system. You can use iBGP for multihomed BGP networks (networks that have more than one connection to the same external autonomous system).

The figure shows an iBGP network within a larger BGP network.

Figure 16: iBGP Network



iBGP networks are fully meshed. Each iBGP peer has a direct connection to all other iBGP peers to prevent network loops.

For single-hop iBGP peers with update-source configured under neighbor configuration mode, the peer supports fast external fall-over.

You should use loopback interfaces for establishing iBGP peering sessions because loopback interfaces are less susceptible to interface flapping. An interface flap occurs when the interface is administratively brought up or down because of a failure or maintenance issue. See the Configuring eBGP, on page 310 section for information on multihop, fast external fallovers, and limiting the size of the AS_path attribute.

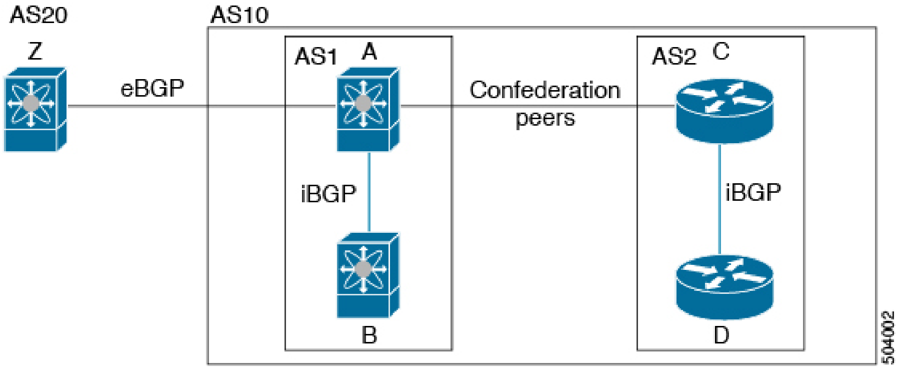
Note You should configure a separate interior gateway protocol in the iBGP network.

AS Confederations

A fully meshed iBGP network becomes complex as the number of iBGP peers grows. You can reduce the iBGP mesh by dividing the autonomous system into multiple subautonomous systems and grouping them into a single confederation. A confederation is a group of iBGP peers that use the same autonomous system number to communicate to external networks. Each subautonomous system is fully meshed within itself and has a few connections to other subautonomous systems in the same confederation.

The figure shows the BGP network, split into two subautonomous systems and one confederation.

Figure 17: AS Confederation



In this example, AS10 is split into two subautonomous systems, AS1 and AS2. Each subautonomous system is fully meshed, but there is only one link between the subautonomous systems. By using AS confederations, you can reduce the number of links compared to the fully meshed autonomous system.

Route Reflector

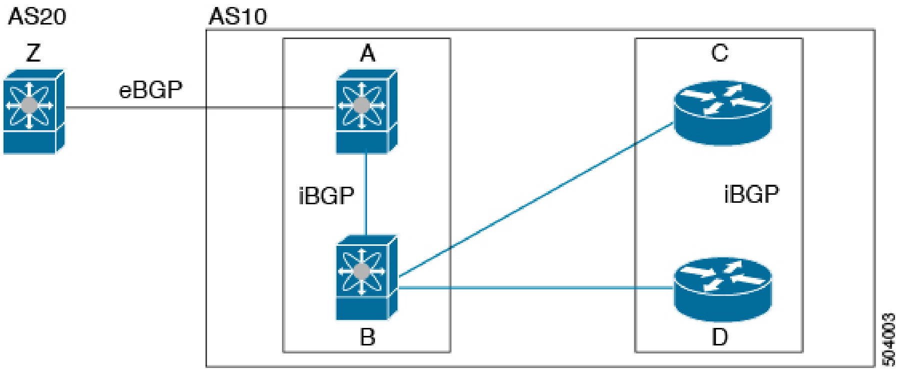
You can alternately reduce the iBGP mesh by using a route reflector configuration where route reflectors pass learned routes to neighbors so that all iBGP peers do not need to be fully meshed.

When you configure an iBGP peer to be a route reflector, it becomes responsible for passing iBGP learned routes to a set of iBGP neighbors.

The figure shows a simple iBGP configuration with four meshed iBGP speakers (routers A, B, C, and D). Without route reflectors, when router A receives a route from an external neighbor, it advertises the route to all three iBGP neighbors.

In the figure, router B is the route reflector. When the route reflector receives routes advertised from router A, it advertises (reflects) the routes to routers C and D. Router A no longer has to advertise to both routers C and D.

Figure 18: Route Reflector



The route reflector and its client peers form a cluster. You do not have to configure all iBGP peers to act as client peers of the route reflector. You must configure any nonclient peer as fully meshed to guarantee that complete BGP updates reach all peers.

Capabilities Negotiation

A BGP speaker can learn about BGP extensions that are supported by a peer by using the capabilities negotiation feature. Capabilities negotiation allows BGP to use only the set of features supported by both BGP peers on a link.

If a BGP peer does not support capabilities negotiation, Cisco NX-OS attempts a new session to the peer without capabilities negotiation if you have configured the address family as IPv4.

Route Dampening

Route dampening is a BGP feature that minimizes the propagation of flapping routes across an internetwork. A route flaps when it alternates between the available and unavailable states in rapid succession.

For example, consider a network with three BGP autonomous systems: AS1, AS2, and AS3. Suppose that a route in AS1 flaps (it becomes unavailable). Without route dampening, AS1 sends a withdraw message to AS2. AS2 propagates the withdrawal message to AS3. When the flapping route reappears, AS1 sends an advertisement message to AS2, which sends the advertisement to AS3. If the route repeatedly becomes unavailable, and then available, AS1 sends many withdrawal and advertisement messages that propagate through the other autonomous systems.

Route dampening can minimize flapping. Suppose that the route flaps. AS2 (in which route dampening is enabled) assigns the route a penalty of 1000. AS2 continues to advertise the status of the route to neighbors. Each time that the route flaps, AS2 adds to the penalty value. When the route flaps so often that the penalty exceeds a configurable suppression limit, AS2 stops advertising the route, regardless of how many times that it flaps. The route is now dampened.

The penalty placed on the route decays until the reuse limit is reached. At that time, AS2 advertises the route again. When the reuse limit is at 50 percent, AS2 removes the dampening information for the route.

Note The router does not apply a penalty to a resetting BGP peer when route dampening is enabled, even though the peer reset withdraws the route.

BGP Best-Path Selection

The BGP best-path algorithm considers the paths as equal-cost paths if the following attributes are identical:

- Weight
- Local preference
- AS_path
- Origin code
- Multi-exit discriminator (MED)
- IGP cost to the BGP next hop

BGP selects only one of these multiple paths as the best path and advertises the path to the BGP peers. For more information, see the BGP Additional Paths, on page 289 section.

Note Paths that are received from different AS confederations are considered as equal-cost paths if the external AS_path values and the other attributes are identical.

Note When you configure a route reflector for iBGP multipath, and the route reflector advertises the selected best path to its peers, the next hop for the path is not modified.

BGP Additional Paths

Only one BGP best path is advertised, and the BGP speaker accepts only one path for a given prefix from a given peer. If a BGP speaker receives multiple paths for the same prefix within the same session, it uses the most recent advertisement.

BGP supports the additional paths feature, which allows the BGP speaker to propagate and accept multiple paths for the same prefix without the new paths replacing any previous ones. This feature allows BGP speaker peers to negotiate whether they support advertising and receiving multiple paths per prefix and advertising such paths. A special 4-byte path ID is added to the network layer reachability information (NLRI) to differentiate multiple paths for the same prefix sent across a peer session. The following figure illustrates the BGP additional paths capability.

Figure 19: BGP Route Advertisement with the Additional Paths Capability

For information on configuring BGP additional paths, see the Configuring BGP Additional Paths, on page 307 section.

Note Cisco Nexus 3550-T hardware does not install ECMP routes.

Route Aggregation

You can configure aggregate addresses. Route aggregation simplifies route tables by replacing a number of more specific addresses with an address that represents all the specific addresses. For example, you can replace these three more specific addresses, 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one aggregate address, 10.1.0.0/16.

Aggregate prefixes are present in the BGP route table so that fewer routes are advertised.

Note Cisco NX-OS does not support automatic route aggregation.

Route aggregation can lead to forwarding loops. To avoid this problem, when BGP generates an advertisement for an aggregate address, it automatically installs a summary discard route for that aggregate address in the local routing table. BGP sets the administrative distance of the summary discard to 220 and sets the route type to discard. BGP does not use discard routes for next-hop resolution.

A summary entry is created in the BGP table when you issue the **aggregate-address** command, but the summary entry is not eligible for advertisement until a subset of the aggregate is found in the table.

BGP Conditional Advertisement

BGP conditional advertisement allows you to configure BGP to advertise or withdraw a route based on whether or not a prefix exists in the BGP table. This feature is useful, for example, in multihomed networks, in which you want BGP to advertise some prefixes to one of the providers only if information from the other provider is not present.

Consider an example network with three BGP autonomous systems: AS1, AS2, and AS3, where AS1 and AS3 connect to the Internet and to AS2. Without conditional advertisement, AS2 propagates all routes to both AS1 and AS3. With conditional advertisement, you can configure AS2 to advertise certain routes to AS3 only if routes from AS1 do not exist (if for example, the link to AS1 fails).

BGP conditional advertisement adds an exist or not-exist test to each route that matches the configured route map. See the Configuring BGP Conditional Advertisement, on page 320 section for more information.

BGP Next-Hop Address Tracking

BGP monitors the next-hop address of installed routes to verify next-hop reachability and to select, install, and validate the BGP best path. BGP next-hop address tracking speeds up this next-hop reachability test by triggering the verification process when routes change in the Routing Information Base (RIB) that may affect BGP next-hop reachability.

BGP receives notifications from the RIB when the next-hop information changes (event-driven notifications). BGP is notified when any of the following events occurs:

- The next hop becomes unreachable.
- The next hop becomes reachable.
- The fully recursed Interior Gateway Protocol (IGP) metric to the next hop changes.
- The first hop IP address or first hop interface changes.
- The next hop becomes connected.

- The next hop becomes unconnected.
- The next hop becomes a local address.
- The next hop becomes a nonlocal address.

Note Reachability and recursed metric events trigger a best-path recalculation.

Event notifications from the RIB are classified as critical and noncritical. Notifications for critical and noncritical events are sent in separate batches. However, a noncritical event is sent with the critical events if the noncritical event is pending and there is a request to read the critical events.

- Critical events are related to next-hop reachability, such as the loss of next hops resulting in a switchover to a different path. A change in the IGP metric for a next hop resulting in a switchover to a different path can also be considered a critical event.
- Non-critical events are related to next hops being added without affecting the best path or changing the IGP metric to a single next hop.

See the Configuring BGP Next-Hop Address Tracking, on page 304 section for more information.

Route Redistribution

You can configure BGP to redistribute static routes or routes from other protocols. You must configure a route map with the redistribution to control which routes are passed into BGP. A route map allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on. See the Configuring Route Policy Manager section, for more information.

You can use route maps to override the default behavior in both scenarios, but be careful when doing so as incorrect use of route maps can result in network loops. The following examples show how to use route maps to change the default behavior.

You can change the default behavior for scenario 1 by modifying the route map as follows:

```
route-map foo permit 10
   match route-type internal
router ospf 1
   redistribute bgp 100 route-map foo
```

Similarly, you can change the default behavior for scenario 2 by modifying the route map as follows:

```
route-map foo deny 10
  match route-type internal
router ospf 1
   vrf bar
     redistribute bgp 100 route-map foo
```

Tuning BGP

You can modify the default behavior of BGP through BGP timers and by adjusting the best-path algorithm.

BGP Timers

BGP uses different types of timers for neighbor session and global protocol events. Each established session has a minimum of two timers for sending periodic keepalive messages and for timing out sessions when peer keepalives do not arrive within the expected time. In addition, there are other timers for handling specific features. Typically, you configure these timers in seconds. The timers include a random adjustment so that the same timers on different BGP peers trigger at different times.

Tuning the Best-Path Algorithm

You can modify the default behavior of the best-path algorithm through optional configuration parameters, including changing how the algorithm handles the multi-exit discriminator (MED) attribute and the router ID.

Graceful Restart and High Availability

Cisco NX-OS supports nonstop forwarding and graceful restart for BGP.

If a Cisco NX-OS router experiences a cold reboot, the network does not forward traffic to the router and removes the router from the network topology. In this scenario, BGP experiences a nongraceful restart and removes all routes. When Cisco NX-OS applies the startup configuration, BGP reestablishes peering sessions and relearns the routes.

When a router detects that a graceful restart operation is in progress, both routers exchange their topology tables. When the router has route updates from all BGP peers, it removes all the stale routes and runs the best-path algorithm on the updated routes.

For single-hop iBGP peers with update-source configured under neighbor configuration mode, the peer supports fast external fall-over.

With the additional BGP paths feature, if the number of paths advertised for a given prefix is the same before and after restart, the choice of path ID guarantees the final state and removal of stale paths. If fewer paths are advertised for a given prefix after a restart, stale paths can occur on the graceful restart helper peer.

Low Memory Handling

BGP reacts to low memory for the following conditions:

- Minor alert—BGP does not establish any new eBGP peers. BGP continues to establish new iBGP peers and confederate peers. Established peers remain, but reset peers are not re-established.
- Severe alert—BGP shuts down select established eBGP peers every two minutes until the memory alert becomes minor. For each eBGP peer, BGP calculates the ratio of total number of paths received to the number of paths selected as best paths. The peers with the highest ratio are selected to be shut down to reduce memory usage. You must clear a shutdown eBGP peer before you can bring the eBGP peer back up to avoid oscillation.

Note You can exempt important eBGP peers from this selection process.

- Critical alert—BGP gracefully shuts down all the established peers. You must clear a shutdown BGP peer before you can bring the BGP peer back up.

See the Tuning BGP , on page 291 section for more information on how to exempt a BGP peer from a shutdown due to a low memory condition.

Prerequisites for Advanced BGP

Advanced BGP has the following prerequisites:

- You must enable BGP (see the Enabling BGP section).
- You should have a valid router ID configured on the system.
- You must have an AS number, either assigned by a Regional Internet Registry (RIR) or locally administered.
- You must have reachability (such as an interior gateway protocol [IGP], a static route, or a direct connection) to the peer that you are trying to make a neighbor relationship with.
- You must explicitly configure an address family under a neighbor for the BGP session establishment.

Guidelines and Limitations for Advanced BGP

Note ***Cisco Nexus 3550-T - 10.1(2t) release***, BGP supports only default VRF.

Advanced BGP has the following configuration guidelines and limitations:

- Prefix peering operates only in passive TCP mode. It accepts incoming connections from remote peers if the peer address falls within the prefix.
- Configuring the **advertise-maps** command multiple times is not supported.
- The dynamic AS number prefix peer configuration overrides the individual AS number configuration that is inherited from a BGP template.
- If you configure a dynamic AS number for prefix peers in an AS confederation, BGP establishes sessions with only the AS numbers in the local confederation.
- BGP sessions that are created through a dynamic AS number prefix peer ignore any configured eBGP multihop time-to-live (TTL) value or a disabled check for directly connected peers.
- Configure a router ID for BGP to avoid automatic router ID changes and session flaps.
- Use the maximum-prefix configuration option per peer to restrict the number of routes that are received and system resources used.
- Configure the update source to establish a session with eBGP multihop sessions.
- Specify a BGP route map if you configure a redistribution.
- Configure the BGP router ID within a VRF.

Note Only 48 BGP sessions are validated in Cisco Nexus 3550-T.

- If you decrease the keepalive and hold timer values, the network might experience session flaps.
- When you redistribute BGP to IGP, iBGP is redistributed as well. To override this behavior, you must insert an extra deny statement into the route map.
- The following guidelines and limitations apply to the **remove-private-as** command:
    - It applies only to eBGP peers.
    - It can be configured only in neighbor configuration mode and not in neighbor-address-family mode.
    - If the AS-path includes both private and public AS numbers, the private AS numbers are not removed.
    - If the AS-path contains the AS number of the eBGP neighbor, the private AS numbers are not removed.
    - Private AS numbers are removed only if all AS numbers in that AS-path belong to a private AS number range. Private AS numbers are not removed if a peer's AS number or a non-private AS number is found in the AS-path segment.
- If you disable a command in the neighbor, template peer, template peer-session, or template peer-policy configuration mode (and the **inherit peer** or **inherit peer-session** command is present), you must use the **default** keyword to return the command to its default state. For example, to disable the **update-source loopback 0** command from the running configuration, you must enter the **default update-source loopback 0** command.
- When next-hop-self is configured for route-reflector clients, the route reflector advertises routes to its clients with itself as the next hop.

Default Settings

The table lists the default settings for advanced BGP parameters.

Parameters	Default
BGP feature	Disabled
BGP additional paths	Disabled
Keep alive interval	60 seconds
Hold timer	180 seconds
Dynamic capability	Enabled

Configuring BGP Session Templates

You can use BGP session templates to simplify the BGP configuration for multiple BGP peers with similar configuration needs. BGP templates allow you to reuse common configuration blocks. You configure BGP templates first and then apply these templates to BGP peers.

With BGP session templates, you can configure session attributes such as inheritance, passwords, timers, and security.

A peer-session template can inherit from one other peer-session template. You can configure the second template to inherit from a third template. The first template also inherits this third template. This indirect inheritance can continue for up to seven peer-session templates.

Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template.

Before you begin

You must enable BGP (see the Enabling BGP section).

Note When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the default form of the command to reset that attribute to the default state.

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		Enables BGP and assigns the autonomous system number to the local BGP speaker.

Step 3	**template peer-session** *template-name*
	Example:
	`switch(config-router)# template peer-session BaseSession`
	`switch(config-router-stmp)#`
		Enters peer-session template configuration mode.

Step 4	(Optional) **password** *number password* **Example:** `switch(config-router-stmp)# password 0 test`	Adds the clear text password test to the neighbor. The password is stored and displayed in type 3 encrypted form (3DES).

	Command or Action	Purpose
Step 5	(Optional) **timers** *keepalive hold* **Example:** `switch(config-router-stmp)# timers 30 90`	Adds the BGP keepalive and holdtimer values to the peer-session template. The default keepalive interval is 60. The default hold time is 180.

Step 6	**exit**
	Example:
	`switch(config-router-stmp)# exit`
	`switch(config-router)#`
		Exits peer-session template configuration mode.

Step 7	**neighbor** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65535`
	`switch(config-router-neighbor)#`
		Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.

Step 8	**inherit peer-session** *template-name*
	Example:
	`switch(config-router-neighbor)# inherit peer-session`
	`BaseSession`
	`switch(config-router-neighbor)#`
		Applies a peer-session template to the peer.

Step 9	(Optional) **description** *text*
	Example:
	`switch(config-router-neighbor)#`
	`description Peer Router A`
	`switch(config-router-neighbor)#`
		Adds a description for the neighbor.

Step 10	(Optional) **show bgp peer-session** *template-name*
	Example:
	`switch(config-router-neighbor)# show bgp`
	`peer-session BaseSession`
		Displays the peer-policy template.

Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor)# copy running-config startup-config`	Saves this configuration change. Use the **show bgp neighbor** command to see the template applied.

Example

This example shows how to configure a BGP peer-session template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BaseSession
switch(config-router-stmp)# timers 30 90
switch(config-router-stmp)# exit
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# description Peer Router A
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring BGP Peer-Policy Templates

You can configure a peer-policy template to define attributes for a particular address family. You assign a preference to each peer-policy template and these templates are inherited in the order specified, for up to five peer-policy templates in a neighbor address family.

Cisco NX-OS evaluates multiple peer policies for an address family using the preference value. The lowest preference value is evaluated first. Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template.

Peer-policy templates can configure address family-specific attributes such as AS-path filter lists, prefix lists, route reflection, and soft reconfiguration.

Note Use the **show bgp neighbor** command to see the template applied.

Before you begin

You must enable BGP (see the Enabling BGP section).

Note When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the default form of the command to reset that attribute to the default state.

Procedure

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

Step 2	**router bgp** *autonomous-system-number*
	Example:
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		Enables BGP and assigns the autonomous system number to the local BGP speaker.

	Command or Action	Purpose

Step 3	**template peer-session** *template-name*
	Example:
	`switch(config-router)# template peer-policy BasePolicy`
	`switch(config-router-ptmp)#`
		Creates a peer-policy template.

Step 4	(Optional) **advertise-active-only** **Example:** `switch(config-router-ptmp)# advertise-active-only`	Advertises only active routes to the peer.
Step 5	(Optional) **maximum-prefix** *number* **Example:** `switch(config-router-ptmp)# maximum-prefix 20`	Sets the maximum number of prefixes allowed from this peer.

Step 6	**exit**
	Example:
	`switch(config-router-ptmp)# exit`
	`switch(config-router)#`
		Exits peer-policy template configuration mode.

Step 7	**neighbor** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65535`
	`switch(config-router-neighbor)#`
		Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.

Step 8	**address-family** {**ipv4**} {**unicast**}
	Example:
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Enters global address family configuration mode for the address family specified.

Step 9	**inherit peer-policy** *template-name preference* **Example:** `switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1`	Applies a peer-policy template to the peer address family configuration and assigns the preference value for this peer policy.
Step 10	(Optional) **show bgp peer-policy** *template-name* **Example:** `switch(config-router-neighbor-af)# show bgp peer-policy BasePolicy`	Displays the peer-policy template.
Step 11	(Optional) **copy running-config startup-config** **Example:**	Saves this configuration change. Use the **show bgp neighbor** command to see the template applied.

	Command or Action	Purpose
	`switch(config-router-neighbor-af)# copy running-config startup-config`	

Example

This example shows how to configure a BGP peer-policy template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BasePolicy
switch(config-router-ptmp)# maximum-prefix 20
switch(config-router-ptmp)# exit
switch(config-router)# neighbor 192.168.1.1 remote-as 65536
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring BGP Peer Templates

You can configure BGP peer templates to combine session and policy attributes in one reusable configuration block. Peer templates can also inherit peer-session or peer-policy templates. Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template. You configure only one peer template for a neighbor, but that peer template can inherit peer-session and peer-policy templates.

Peer templates support session and address family attributes, such as eBGP multihop time-to-live, maximum prefix, next-hop self, and timers.

Before you begin

You must enable BGP (see the Enabling BGP section).

Note When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the default form of the command to reset that attribute to the default state.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system-number* **Example:** `switch(config)# router bgp 65535`	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.

	Command or Action	Purpose
Step 3	**template peer** *template-name* **Example:** `switch(config-router)# template peer BasePeer`	Enters peer template configuration mode.
Step 4	(Optional) **inherit peer-session** *template-name* **Example:** `switch(config-router-neighbor)# inherit peer-session BaseSession`	Adds a peer-session template to the peer template.

Step 5	(Optional) **address-family** {**ipv4**} {**unicast**}
	Example:
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)`
		Configures the global address family configuration mode for the specified address family.

Step 6	(Optional) **inherit peer-policy** *template-name* **Example:** `switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1`	Applies a peer-policy template to the neighbor address family configuration.
Step 7	**exit** **Example:** `switch(config-router-neighbor-af)# exit`	Exits BGP neighbor address family configuration mode.
Step 8	(Optional) **timers** *keepalive hold* **Example:** `switch(config-router-neighbor)# timers 45 100`	Adds the BGP timer values to the peer. These values override the timer values in the peer-session template, BaseSession.
Step 9	**exit** **Example:** `switch(config-router-neighbor)# exit`	Exits BGP neighbor configuration mode.

Step 10	**neighbor** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65535`
	`switch(config-router-neighbor)#`
		Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.

Step 11	**inherit peer** *template-name* **Example:** `switch(config-router-neighbor)# inherit peer BasePeer`	Inherits the peer template.
Step 12	(Optional) **timers** *keepalive hold* **Example:**	Adds the BGP timer values to this neighbor.

	Command or Action	Purpose
	`switch(config-router-neighbor)# timers 60 120`	These values override the timer values in the peer template and the peer-session template.
Step 13	(Optional) **show bgp peer-template** *template-name* **Example:** `switch(config-router-neighbor)# show bgp peer-template BasePeer`	Displays the peer template.
Step 14	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor)# copy running-config startup-config`	Saves this configuration change. Use the **show bgp neighbor** command to see the template applied.

Example

This example shows how to configure a BGP peer template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer BasePeer
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer BasePeer
switch(config-router-neighbor)# copy running-config startup-config
```

Configuring Prefix Peering

BGP supports the definition of a set of peers using a prefix for both IPv4. This feature allows you to not have to add each neighbor to the configuration.

When defining a prefix peering, you must specify the remote AS number with the prefix. BGP accepts any peer that connects from that prefix and autonomous system if the prefix peering does not exceed the configured maximum peers allowed.

When a BGP peer that is part of a prefix peering disconnects, Cisco NX-OS holds its peer structures for a defined prefix peer timeout value. An established peer can reset and reconnect without danger of being blocked because other peers have consumed all slots for that prefix peering.

Procedure

	Command or Action	Purpose
Step 1	**timers prefix-peer-timeout** *value* **Example:** `switch(config-router-neighbor)# timers prefix-peer-timeout 120`	Configures the BGP prefix peering timeout value in router configuration mode. The range is from 0 to 1200 seconds. The default value is 30. **Note** For prefix peers, set the prefix peer timeout to be greater than the configured graceful restart timer. If the prefix peer timeout is greater than the graceful restart timer, a peer's route is retained during its restart. If the prefix peer timeout is less than the graceful restart timer, the peer's route is purged by the prefix peer timeout, which may occur before the restart is complete.
Step 2	**maximum-peers** *value* **Example:** `switch(config-router-neighbor)# maximum-peers 120`	Configures the maximum number of peers for this prefix peering in neighbor configuration mode. The range is from 1 to 1000.

Example

This example shows how to configure a prefix peering that accepts up to 10 peers:

```
switch(config)# router bgp 65536
switch(config-router)# timers prefix-peer-timeout 120
switch(config-router)# neighbor 10.100.200.0/24 remote-as 65536
switch(config-router-neighbor)# maximum-peers 10
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

Use the **show bgp ipv4 unicast neighbors** command to show the details of the configuration for that prefix peering with a list of the currently accepted instances and the counts of active, maximum concurrent, and total accepted peers.

Configuring BGP Authentication

You can configure BGP to authenticate route updates from peers using MD5 digests.

To configure BGP to use MD5 digests, use the following command in neighbor configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**password** {**0** | **3** | **7**} *string* **Example:** `switch(config-router-neighbor)# password BGPpassword`	Configures an MD5 password for BGP neighbor sessions.

Resetting a BGP Session

If you modify a route policy for BGP, you must reset the associated BGP peer sessions. If the BGP peers do not support route refresh, you can configure a soft reconfiguration for inbound policy changes. Cisco NX-OS automatically attempts a soft reset for the session.

To configure soft reconfiguration inbound, use the following command in neighbor address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**soft-reconfiguration inbound** **Example:** `switch(config-router-neighbor-af)# soft-reconfiguration inbound`	Enables soft reconfiguration to store the inbound BGP route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

Step 2	(Optional) **clear bgp** {**ipv4** } {**unicast** *ip-address* **soft** {**in** | **out**}
	Example:
	`switch# clear bgp ip unicast 192.0.2.1 soft in`
		Resets the BGP session without tearing down the TCP session.

Modifying the Next-Hop Address

You can modify the next-hop address used in a route advertisement in the following ways:

- Disable next-hop calculation and use the local BGP speaker address as the next-hop address.
- Set the next-hop address as a third-party address. Use this feature in situations where the original next-hop address is on the same subnet as the peer that the route is being sent to. Using this feature saves an extra hop during forwarding.

To modify the next-hop address, use the following commands in address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**next-hop-self** **Example:** `switch(config-router-neighbor-af)# next-hop-self`	Uses the local BGP speaker address as the next-hop address in route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 2	**next-hop-third-party** **Example:** `switch(config-router-neighbor-af)# next-hop-third-party`	Sets the next-hop address as a third-party address. Use this command for single-hop eBGP peers that do not have **next-hop-self** configured.

Configuring BGP Next-Hop Address Tracking

BGP next-hop address tracking is enabled by default and cannot be disabled.

You can modify the delay interval between RIB checks to increase the performance of BGP next-hop tracking.

To modify the BGP next-hop address tracking, use the following commands in address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**nexthop trigger-delay** {**critical** | **non-critical**} *milliseconds* **Example:** `switch(config-router-af)# nexthop trigger-delay critical 5000`	Specifies the next-hop address tracking delay timer for critical next-hop reachability routes and for noncritical routes. The range is from 1 to 4294967295 milliseconds. The critical timer default is 3000. The noncritical timer default is 10000.

Configuring Next-Hop Filtering

BGP next-hop filtering allows you to specify that when a next-hop address is checked with the RIB, the underlying route for that next-hop address is passed through the route map. If the route map rejects the route, the next-hop address is treated as unreachable.

BGP marks all next hops that are rejected by the route policy as invalid and does not calculate the best path for the routes that use the invalid next-hop address.

To configure BGP next-hop filtering, use the following command in address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**nexthop route-map** *name* **Example:** `switch(config-router-af)# nexthop route-map nextHopLimits`	Specifies a route map to match the BGP next-hop route to. The name can be any case-sensitive, alphanumeric string up to 63 characters.

Configuring Next-Hop Resolution via Default Route

BGP next-hop resolution allows you to specify if the IP default route is used for BGP next-hop resolution.

To configure BGP next-hop resolution, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	[**no**] **nexthop suppress-default-resolution** **Example:** `switch(config-router)# nexthop suppress-default-resolution`	Prevents resolution of BGP next hop through the IP default route. When this command is enabled: • The output of the **show bgp process detail** command includes the following line: Use default route for nexthop resolution: No • The output of the **show routing clients bgp** command includes the following line: Owned rnh will never resolve to 0.0.0.0/0

Controlling Reflected Routes Through Next-Hop-Self

NX-OS enables controlling the iBGP routes being sent to a specific peer through the **next-hop-self** [all] arguments. By using these arguments, you can selectively change the next-hop of routes even if the route is reflected.

Command	Purpose
next-hop-self [all] **Example**: `switch(config-router-af)# next-hop-self all`	Uses the local BGP speaker address as the next-hop address in route updates. The all keyword is optional. If you specify all, all routes are sent to the peer with next-hop-self. If you do not specify all, the next hops of reflected routes are not changed.

Shrinking Next-Hop Groups When A Session Goes Down

This feature applies to the following BGP path failure events:

- Any single or multiple Layer 3 link failures
- Line card failures
- Administrative shutdown of BGP neighbors (using the shutdown command)

The accelerated handling of the first two events (Layer 3 link failures and line card failures) is enabled by default and does not require a configuration command to be enabled.

To configure the accelerated handling of the last two events, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**neighbor-down fib-accelerate** **Example:** `switch(config-router)# neighbor-down fib-accelerate`	Withdraws the corresponding next hop from all next-hop groups (single next-hop routes) whenever a BGP session goes down. **Note** This command applies to both IPv4 routes.

Disabling Capabilities Negotiation

You can disable capabilities negotiations to interoperate with older BGP peers that do not support capabilities negotiation.

To disable capabilities negotiation, use the following command in neighbor configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**dont-capability-negotiate** **Example:** `switch(config-router-neighbor)# dont-capability-negotiate`	Disables capabilities negotiation. You must manually reset the BGP sessions after configuring this command.

Disabling Policy Batching

In BGP deployments where prefixes have unique attributes, BGP tries to identify routes with similar attributes to bundle in the same BGP update message. To avoid the overhead of this additional BGP processing, you can disable batching.

Cisco recommends that you disable policy batching for BGP deployments that have a large number of routes with unique next hops.

To disable policy batching, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**disable-policy-batching** **Example:** `switch(config-router)# disable-policy-batching`	Disables the batching evaluation of prefix advertisements to all peers.

Configuring BGP Additional Paths

BGP supports sending and receiving multiple paths per prefix and advertising such paths.

Advertising the Capability of Sending and Receiving Additional Paths

You can configure BGP to advertise the capability of sending and receiving additional paths to and from the BGP peers. To do so, use the following commands in neighbor address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	[**no**] **capability additional-paths send** [**disable**] **Example:** `switch(config-router-neighbor-af)# capability addtional-paths send`	Advertises the capability to send additional paths to the BGP peer. The **disable** option disables the advertising capability of sending additional paths. The **no** form of this command disables the capability of sending additional paths.
Step 2	[**no**] **capability additional-paths receive** [**disable**] **Example:** `switch(config-router-neighbor-af)# capability addtional-paths receive`	Advertises the capability to receive additional paths from the BGP peer. The **disable** option disables the advertising capability of receiving additional paths. The **no** form of this command disables the capability of receiving additional paths.
Step 3	**show bgp neighbor** **Example:** `switch(config-router-neighbor-af)# show bgp neighbor`	Displays whether the local peer has advertised the additional paths send or receive capability to the remote peer.

Example

This example shows how to configure BGP to advertise the capability to send and receive additional paths to and from the BGP peer:

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# neighbor 10.131.31.2 remote-as 100
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# capability additional-paths send
switch(config-router-neighbor-af)# capability additional-paths receive
```

Configuring the Sending and Receiving of Additional Paths

You can configure the capability of sending and receiving additional paths to and from the BGP peers. To do so, use the following commands in address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	[**no**] **additional-paths send** **Example:** `switch(config-router-af)# additional-paths send`	Enables the send capability of additional paths for all of the neighbors under this address family for which the capability has not been disabled. The **no** form of this command disables the send capability.
Step 2	[**no**] **additional-paths receive** **Example:** `switch(config-router-af)# additional-paths receive`	Enables the receive capability of additional paths for all of the neighbors under this address family for which the capability has not been disabled. The **no** form of this command disables the receive capability.
Step 3	**show bgp neighbor** **Example:** `switch(config-router-af)# show bgp neighbor`	Displays whether the local peer as advertised the additional paths send or receive capability to the remote peer.

Example

This example shows how to enable the additional paths send and receive capability for all neighbors under the specified address family for which this capability has not been disabled:

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# additional-paths send
switch(config-router-af)# additional-paths receive
```

Configuring Advertised Paths

You can specify the paths that are advertised for BGP. To do so, use the following commands in route-map configuration mode:

Procedure

	Command or Action	Purpose
Step 1	[**no**] **set ip next-hop unchanged** **Example:** `switch(config-route-map)# set ip next-hop unchanged`	Specifies and unchanged next-hop IP address.
Step 2	[**no**] **set path-selection { all | backup | best2}** | advertise **Example:** `switch(config-route-map)# set path-selection all advertise`	Specifies that all paths be advertised for a given prefix. You can use one of the following options: • all—Advertises all available valid paths. • backup—Advertises paths marked as backup paths. This option requires that backup paths be enabled using the additional-path install backup command. • best2—Advertises the second best path, which is the best path of the remaining available paths, except the already calculated best path. The **no** form of this command specifies that only the best path be advertised.
Step 3	**show bgp** {**ipv4** } **unicast** [*ip-address*] **Example:** `switch(config-route-map)# show bgp ipv4 unicast`	Displays the path ID for the additional paths of a prefix and advertisement information for these paths.

Example

This example show how to specify that all paths be advertised for the prefix list p1:

```
switch# configure terminal
switch(config)# route-map PATH_SELECTION_RMAP
switch(config-route-map)# match ip address prefix-list p1
switch(config-route-map)# set path-selection all advertise
```

Configuring Additional Path Selection

You can configure the capability fo selecting additional paths for a prefix. To do so, use the following commands in address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	[**no**] **additional-paths selection route-map** *map-name* **Example:** `switch(config-router-af)# additional paths selection route-map map1`	Configures the capability of selecting additional paths for a prefix. The **no** form of this command disables the additional paths selection capability.
Step 2	**show bgp** {**ipv4** } **unicast** [*ip-address*] **Example:** `switch(config-route-af)# show bgp ipv4 unicast`	Displays the path ID for the additional paths of a prefix and advertisement information for these paths.

Example

This example shows how to configure additional paths selection under the specified address family:

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# additional-paths selection route-map PATH_SELECTION_RMAP
```

Configuring eBGP

Disabling eBGP Single-Hop Checking

You can configure eBGP to disable checking whether a single-hop eBGP peer is directly connected to the local router. Use this option for configuring a single-hop loopback eBGP session between directly connected switches.

To disable checking whether or not a single-hop eBGP peer is directly connected, use the following command in neighbor configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**disable-connected-check** **Example:** `switch(config-router-neighbor)# disable-connected-check`	Disables checking whether or not a single-hop eBGP peer is directly connected. You must manually reset the BGP sessions after using this command.

Configuring eBGP Multihop

You can configure the eBGP time-to-live (TTL) value to support eBGP multihop. In some situations, an eBGP peer is not directly connected to another eBGP peer and requires multiple hops to reach the remote eBGP peer. You can configure the eBGP TTL value for a neighbor session to allow these multihop sessions.

Note This configuration is not supported for BGP interface peering.

To configure eBGP multihop, use the following command in neighbor configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**ebgp-multihop** *ttl-value* **Example:** `switch(config-router-neighbor)# ebgp-multihop 5`	Configures the eBGP TTL value for eBGP multihop. The range is from 2 to 255. You must manually reset the BGP sessions after using this command.

Disabling a Fast External Fallover

Be default, the Cisco NX-OS device supports fast external fallover for neighbors in all VRFs and address families (IPv4). Typically, when a BGP router loses connectivity to a directly connected eBGP peer, BGP triggers a fast external fallover by resetting the eBGP session to the peer. You can disable this fast external fallover to limit the instability caused by link flaps.

To disable fast external fallover, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**no fast-external-fallover** **Example:** `switch(config-router)# no fast-external-fallover`	Disables a fast external fallover for eBGP peers. This command is enabled by default.

Limiting the AS-path Attribute

You can configure eBGP to discard routes that have a high number of AS numbers in the AS-path attribute.

To discard routes that have a high number of AS numbers in the AS-path attribute, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**maxas-limit** *number* **Example:** `switch(config-router)# maxas-limit 50`	Discards eBGP routes that have a number of AS-path segments that exceed the specified limit. The range is from 1 to 2000.

Configuring Local AS Support

The local-AS feature allows a router to appear to be a member of a second autonomous system (AS), in addition to its real AS. Local AS allows two ISPs to merge without modifying peering arrangements. Routers in the merged ISP become members of the new autonomous system but continue to use their old AS numbers for their customers.

This feature can only be used for true eBGP peers. You cannot use this feature for two peers that are members of different confederation subautonomous systems.

To configure eBGP local AS support, use the following command in neighbor configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**local-as** *number* [**no-prepend** [**replace-as** [**dual-as**]]] **Example:** `switch(config-router-neighbor)# local-as 1.1`	Configures eBGP to prepend the local AS *number* to the AS_PATH attribute. The AS *number* can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Example

This example shows how to configure local AS support on a VRF:

```
switch# configure terminal
switch(config)# router bgp 1
switch(config-router)# neighbor 10.1.1.1
switch(config-router-neighbor)# local-as 1
switch(config-router-neighbor)# show running-config bgp
```

Configuring AS Confederations

To configure an AS confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems within the AS confederation look like a single autonomous system with the confederation identifier as the autonomous system number.

To configure a BGP confederation identifier, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**confederation identifier** *as-number* **Example:** `switch(config-router)# confederation identifier 4000`	In router configuration mode, this command configures a BGP confederation identifier. The command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 2	**bgp confederation peers** *as-number* [*as-number2...*] **Example:** `switch(config-router)# bgp confederation peers 5 33 44`	In router configuration mode, this command configures the autonomous systems that belong to the AS confederation. The command specifies a list of autonomous systems that belong to the confederation and it triggers an automatic notification and session reset for the BGP neighbor sessions.

Configuring Route Reflector

You can configure iBGP peers as route reflector clients to the local BGP speaker, which acts as the route reflector. Together, a route reflector and its clients form a cluster. A cluster of clients usually has a single route reflector. In such instances, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid a single point of failure in the network, you can configure a cluster with more than one route reflector. You must configure all route reflectors in the cluster with the same 4-byte cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster.

Before you begin

You must enable BGP.

Procedure

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

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		Enters BGP mode and assigns the autonomous system number to the local BGP speaker.

Step 3	**cluster-id** *cluster-id* **Example:** `switch(config-router)# cluster-id 192.0.2.1`	Configures the local router as one of the route reflectors that serve the cluster. You specify a cluster ID to identify the cluster. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

	Command or Action	Purpose

Step 4	**address-family** {**ipv4**} {**unicast**}
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Enters router address family configuration mode for the specified address family.

Step 5	(Optional) **client-to-client reflection** **Example:** `switch(config-router-af)# client-to-client reflection`	Configures client-to-client route reflection. This feature is enabled by default. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

Step 6	**exit**
	Example:
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		Exits router address configuration mode.

Step 7	**neighbor** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 192.0.2.10 remote-as 65535`
	`switch(config-router-neighbor)#`
		Configures the IP address and AS number for a remote BGP peer.

Step 8	**address-family** {**ipv4**} {**unicast**}
	Example:
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Enters neighbor address family configuration mode for the unicast IPv4 address family.

Step 9	**route-reflector-client** **Example:** `switch(config-router-neighbor-af)# route-reflector-client`	Configures the device as a BGP route reflector and configures the neighbor as its client. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 10	(Optional) **show bgp** {**ipv4**} {**unicast**} **neighbors** **Example:** `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	Displays the BGP peers.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to configure the router as a route reflector and add one neighbor as a client:

```
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.10 remote-as 65536
switch(config-router-neighbor)# address-family ip unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring Next-Hops on Reflected Routes Using an Outbound Route-Map

You can change the next-hop on reflected routes on a BGP route reflector using an outbound route-map. You can configure the outbound route-map to specify the peer's local address as the next-hop address.

Note The **next-hop-self** command does not enable this functionality for routes being reflected to clients by a route reflector. This functionality can only be enabled using an outbound route-map.

Before you begin

You must enable BGP (see the Enabling BGP section).

You must enter the **set next-hop** command to configure an address family-specific next-hop address.

- When setting IPv4 next-hops using route-maps—If **set ip next-hop peer-address** matches the route-map, the next-hop is set to the peer's local address. If no next-hop is set in the route-map, the next-hop is set to the one stored in the path.

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 200`
	`switch(config-router)#`
		Enters BGP mode and assigns the autonomous system number to the local BGP speaker.

Step 3	**neighbor** *ip-address* **remote-as** *as-number* **Example:**	Configures the IP address and AS number for a remote BGP peer.

	Command or Action	Purpose

	`switch(config-router)# neighbor 192.0.2.12 remote-as 200`
	`switch(config-router-neighbor)#`	

Step 4	(Optional) **update-source** *interface number* **Example:** `switch(config-router-neighbor)# update-source loopback 300`	Specifies and updates the source of the BGP session.

Step 5	**address-family** {**ipv4** } {**unicast**}
	Example:
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Enters router address family configuration mode for the specified address family.

Step 6	**route-reflector-client** **Example:** `switch(config-router-neighbor-af)# route-reflector-client`	Configures the device as a BGP route reflector and configures the neighbor as its client. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 7	**route-map** *map-name* **out** **Example:** `switch(config-router-neighbor-af)# route-map setrrnh out`	Applies the configured BGP policy to outgoing routes.
Step 8	(Optional) **show bgp** {**ipv4** } {**unicast**} [**ip-address**] **route-map** *map-name*] **Example:** `switch(config-router-neighbor-af)# show bgp ipv4 unicast route-map setrrnh`	Displays the BGP routes that match the route map.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to configure the next-hop on reflected routes on a BGP route reflector using an outbound route-map:

```
switch(config)# interface loopback 300
switch(config-if)# ip address 192.0.2.11/32
switch(config-if)# ip router ospf 1 area 0.0.0.0
switch(config-if)# exit
switch(config)# route-map setrrnh permit 10
switch(config-route-map)# set ip next-hop peer-address
switch(config-route-map)# exit
switch(config)# router bgp 200
switch(config-router)# neighbor 192.0.2.12 remote-as 200
switch(config-router-neighbor)# update-source loopback 300
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# route-map setrrnh out
switch(config-router-neighbor-af)# exit
```

Configuring Route Dampening

You can configure route dampening to minimize route flaps propagating through your iBGP network.

To configure route dampening, use the following command in address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**dampening** [{*half-life reuse-limit suppress-limit max-suppress-time* | **route-map** *map-name*}] **Example:** `switch(config-router-af)# dampening route-map bgpDamp`	Disables capabilities negotiation. The parameter values are as follows: • *half-life*—The range is from 1 to 45. • *resuse-limit*—The range is from 1 to 20000. • *suppress-limit*—The range is from 1 to 20000. • *max-suppress-time*—The range is from 1 to 255.

Configuring Maximum Prefixes

You can configure the maximum number of prefixes that BGP can receive from a BGP peer. If the number of prefixes exceeds this value, you can optionally configure BGP to generate a warning message or tear down the BGP session to the peer.

To configure the maximum allowed prefixes for a BGP peer, use the following command in neighbor address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**maximum-prefix** *maximum* [*threshold*] [**restart** *time* | **warning-only**] **Example:** `switch(config-router-neighbor-af)# maximum-prefix 12`	Configures the maximum number of prefixes from a peer. The parameter ranges are as follows: • *maximum*—The range is from 1 to 300000. • *threshold*—The range is from 1 to 100 percent. The default is 75 percent.

	Command or Action	Purpose
		• *time*—The range is from 1 to 65535 minutes. This command triggers an automatic notification and session reset for the BGP neighbor sessions if the prefix is exceeded.

Configuring DSCP

You can configure a differentiated services code point (DSCP) for a neighbor. You can specify a DSCP value for locally originated packets for IPv4.

To configure the DSCP value, use the following command in neighbor configuration mode:

Procedure

	Command or Action	Purpose

Step 1	**dscp** *dscp_value*
	Example:
	`switch(config-router-neighbor)# dscp 63`
	Below is an example of the corresponding **show** command:
	`show ipv4 bgp neighbors`
	`BGP neighbor is 10.1.1.1, remote AS 0, unknown link, Peer index 4`
	`  BGP version 4, remote router ID 0.0.0.0`
	`  BGP state = Idle, down for 00:13:34, retry in 0.000000`
	`  DSCP (DiffServ CodePoint): 0`
	`  Last read never, hold time = 180, keepalive interval is 60 seconds`
		Sets the differentiated services code point (DSCP) value for the neighbor. The DSCP value can be a number from 0 to 63, or it can be one of the following keywords: **ef**, **af11**, **af12**, **af13**, **af21**, **af22**, **af23**, **af31**, **af32**, **af33**, **af41**, **af42**, **af43**, **cs1**, **cs2**, **cs3**, **cs4**, **cs5**, **cs6**, or **cs7**.
		The default value is cs6.
		Note Cisco Nexus 3550-T hardware does not look at the DSCP values in the packets.

Configuring Dynamic Capability

You can configure dynamic capability for a BGP peer.

To configure dynamic capability, use the following command in neighbor configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**dynamic-capability** **Example:** `switch(config-router-neighbor)# dynamic-capability`	Enables dynamic capability. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

Configuring Aggregate Addresses

You can configure aggregate address entries in the BGP route table.

To configure an aggregate address, use the following command in router address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**aggregate-address** *ip-prefix/length* [**as-set**] [**summary-only**] [**advertise-map** *map-name*] [**attribute-map** *map-name*] [**suppress-map** *map-name*] **Example:** `switch(config-router-af)# aggregate-address 192.0.2.0/8 as-set`	Creates an aggregate address. The path advertised for this route is an autonomous system set that consists of all elements contained in all paths that are being summarized: • The **as-set** keyword generates autonomous system set path information and community information from contributing paths. • The **summary-only** keyword filters all more specific routes from updates. • The **advertise-map** keyword and argument specify the route map used to select attribute information from selected routes. • The **attribute-map** keyword and argument specify the route map used to select attribute information from the aggregate. • The **suppress-map** keyword and argument conditionally filter more specific routes. If you specify the **suppress-map** option while performing a BGP route aggregation, you can either suppress certain more-specific routes from being advertised to its peers, or decide to advertise the more-specific routes with some community attributes set on them, depending upon the suppress-map route-map configuration. A route-map configured with only match

	Command or Action	Purpose
		clauses will suppress the more-specific routes that satisfy the match criteria. However, if a route-map is configured with match and set clauses, then the routes satisfying the match criteria will be advertised with the appropriate attributes as modified by the route-map. The second option enables you to set community attributes on the more-specific routes.

Suppressing BGP Routes

You can configure Cisco NX-OS to advertise newly learned BGP routes only after these routes are confirmed by the Forwarding Information Base (FIB) and programmed in the hardware. After the routes are programmed, subsequent changes to these routes do not require this hardware-programming check.

To suppress BGP routes, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**suppress-fib-pending** **Example:** `switch(config-router)# suppress-fib-pending`	Suppresses newly learned BGP routes (IPv4) from being advertised to downstream BGP neighbors until the routes have been programmed in the hardware.

Configuring BGP Conditional Advertisement

You can configure BGP conditional advertisement to limit the routes that BGP propagates. You define the following two route maps:

- Advertise map—Specifies the conditions that the route must match before BGP considers the conditional advertisement. This route map can contain any appropriate match statements.
- Exist map or nonexist map—Defines the prefix that must exist in the BGP table before BGP propagates a route that matches the advertise map. The nonexist map defines the prefix that must not exist in the BGP table before BGP propagates a route that matches the advertise map. BGP processes only the permit statements in the prefix list match statements in these route maps.

If the route does not pass the condition, BGP withdraws the route if it exists in the BGP table.

Before you begin

You must enable BGP (see the Enabling BGP section).

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		Enters BGP mode and assigns the autonomous system number to the local BGP speaker.

Step 3	**neighbor** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65534`
	`switch(config-router-neighbor)#`
		Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.

Step 4	**address-family** {**ipv4**} {**unicast**}
	Example:
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Enters address family configuration mode.

Step 5	**advertise-map** *adv-map* {**exist-map** *exist-rmap*|**non-exist-map** *nonexist-rmap*}
	Example:
	`switch(config-router-neighbor-af)# advertise-map advertise exist-map exist`
		Configures BGP to conditionally advertise routes based on the two configured route maps:
		• *adv-map*—Specifies a route map with **match** statements that the route must pass before BGP passes the route to the next route map. The *adv-map* is a case-sensitive, alphanumeric string up to 63 characters.
		• *exist-rmap*—Specifies a route map with match statements for a prefix list. A prefix in the BGP table must match a prefix in the prefix list before BGP advertises the route. The *exist-rmap* is a case-sensitive, alphanumeric string up to 63 characters.
		• *nonexist-rmap*—Specifies a route map with match statements for a prefix list. A prefix in the BGP table must not match a prefix in the prefix list before BGP advertises the route. The *nonexist-rmap* is a case-sensitive, alphanumeric string up to 63 characters.

	Command or Action	Purpose
Step 6	(Optional) **show bgp** {**ipv4**} {**unicast**} **neighbors** **Example:** `switch(config-router-neighbor-af)# show ip bgp neighbor`	Displays information about BGP and the configured conditional advertisement route maps.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to configure BGP conditional advertisement:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.2 remote-as 65537
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# advertise-map advertise exist-map exist
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
switch(config-router)# exit
switch(config)# route-map advertise
switch(config-route-map)# match as-path pathList
switch(config-route-map)# exit
switch(config)# route-map exit
switch(config-route-map)# match ip address prefix-list plist
switch(config-route-map)# exit
switch(config)# ip prefix-list plist permit 209.165.201.0/27
```

Configuring Route Redistribution

You can configure BGP to accept routing information from another routing protocol and redistribute that information through the BGP network. Optionally, you can assign a default route for redistributed routes.

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** *as-number* **Example:**	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.

	Command or Action	Purpose

	`switch(config)# router bgp 65535`
	`switch(config-router)#`	

Step 3	**address-family ipv4** {**unicast**}
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Enters address family configuration mode.

Step 4	**redistribute** {**direct**| {**ospf** } *instance-tag* | **static**} **route-map** *map-name* **Example:** `switch(config-router-af)# redistribute ospf 201 route-map Ospfmap`	Redistributes routes from other protocols into BGP.
Step 5	(Optional) **default-metric** *value* **Example:** `switch(config-router-af)# default-metric 33`	Generates a default route into BGP.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-af)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to redistribute EIGRP into BGP:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# redistribute ospf 201 route-map Ospfmap
switch(config-router-af)# copy running-config startup-config
```

Advertising the Default Route

You can configure BGP to advertise the default route (network 0.0.0.0).

Before you begin

You must enable BGP (see the Enabling BGP section).

Procedure

	Command or Action	Purpose

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

Step 2	**route-map allow permit**
	Example:
	`switch(config)# route-map allow permit`
	`switch(config-route-map)#`
		Enters router map configuration mode and defines the conditions for redistributing routes.

Step 3	**exit**
	Example:
	`switch(config-route-map)# exit`
	`switch(config)#`
		Exits router map configuration mode.

Step 4	**ip route** *ip-address network-mask* **null** *null-interface-number* **Example:** `switch(config)# ip route 192.0.2.1 255.255.255.0 null 0`	Configures the IP address.

Step 5	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		Enters BGP mode and assigns the AS number to the local BGP speaker.

Step 6	**address-family** {**ipv4**} **unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Enters address-family configuration mode.

Step 7	**default-information originate** **Example:** `switch(config-router-af)# default-information originate`	Advertises the default route.
Step 8	**redistribute static route-map allow** **Example:** `switch(config-router-af)# redistribute static route-map allow`	Redistributes the default route.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-af)# copy running-config startup-config`	Saves this configuration change.

Configuring BGP Attribute Filtering and Error Handling

You can configure BGP attribute filtering and error handling to provide an increased level of security. The following features are available and implemented in the following order:

- **Path attribute treat-as-withdraw:** Allows you to treat-as-withdraw a BGP update from a specific neighbor if the update contains a specified attribute type. The prefixes contained in the update are removed from the routing table.
- **Path attribute discard:** Allows you to remove specific path attributes in a BGP update from a specific neighbor.
- **Enhanced attribute error handling:** Prevents peer sessions from flapping due to a malformed update.

Attribute types 1, 2, 3, 4, 5, 8, 14, 15, and 16 cannot be configured for path attribute treat-as-withdraw and path attribute discard. Attribute type 9 (Originator) and type 10 (Cluster-id) can be configured for eBGP neighbors only.

Treating as Withdraw Path Attributes from a BGP Update Message

To "treat-as-withdraw" BGP updates that contain specific path attributes, use the following command in router neighbor configuration mode:

Procedure

	Command or Action	Purpose

Step 1	[**no**] **path-attribute treat-as-withdraw** [*value* | **range** *start end*] **in**
	Example:
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute treat-as-withdraw 100 in`
	Example:
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute treat-as-withdraw range 21 255 in`
		Treats as withdraw any incoming BGP update messages that contain the specified path attribute or range of path attributes and triggers an inbound route refresh to ensure that the routing table is up to date. Any prefixes in a BGP update that are treat-as-withdraw are removed from the BGP routing table.
		This command is also supported for BGP template peers and BGP template peer sessions.

Discarding Path Attributes from a BGP Update Message

To discard BGP updates that contain specific path attributes, use the following command in router neighbor configuration mode:

Procedure

	Command or Action	Purpose

Step 1	[**no**] **path-attribute discard** [*value* | **range** *start end*] **in**
	Example:
	`switch#(config-router)# neighbor 10.20.30.40 switch(config-router-neighbor)# path-attribute discard 100 in`
	Example:
	`switch#(config-router)# neighbor 10.20.30.40 switch(config-router-neighbor)# path-attribute discard range 100 255 in`
		Drops specified path attributes in BGP update messages for the specified neighbor and triggers an inbound route refresh to ensure that the routing table is up to date. You can configure a specific attribute or an entire range of unwanted attributes.
		This command is also supported for BGP template peers and BGP template peer sessions.
		Note When the same path attribute is configured for both discard and treat-as-withdaw, treat-as-withdraw has a higher priority.

Enabling or Disabling Enhanced Attribute Error Handling

BGP enhanced attribute error handling is enabled by default but can be disabled. This feature, which complies with RFC 7606, prevents peer sessions from flapping due to a malformed update. The default behavior applies to both eBGP and iBGP peers.

To disable or reenable enhanced error handling, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose

Step 1	[**no**] **enhanced-error**
	Example:
	`switch(config)# router bgp 1000`
	`switch(config-router)# enhanced-error`
		Enables or disables BGP enhanced attribute error handling.

Displaying Discarded or Unknown Path Attributes

To display information about discarded or unknown path attributes, perform one of the following tasks:

Command	Purpose
show bgp {**ipv4** } **unicast path-attribute discard**]	Displays all prefixes for which an attribute has been discarded.
show bgp {**ipv4** } **unicast path-attribute unknown**]	Displays all prefixes that have an unknown attribute.
show bgp {**ipv4** } **unicast** *ip-address*	Displays the unknown attributes and discarded attributes associated with a prefix.

The following example shows the prefixes for which an attribute has been discarded:

```
switch# show bgp ipv4 unicast path-attribute discard
Network           Next Hop
1.1.1.1/32        20.1.1.1
1.1.1.2/32        20.1.1.1
1.1.1.3/32        20.1.1.1
```

The following example shows the prefixes that have an unknown attribute:

```
switch# show bgp ipv4 unicast path-attribute unknown
Network           Next Hop
2.2.2.2/32        20.1.1.1
2.2.2.3/32        20.1.1.1
```

The following example shows the unknown attributes and discarded attributes associated with a prefix:

```
switch# show bgp ipv4 unicast 2.2.2.2
BGP routing table entry for 2.2.2.2/32, version 6241
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  1000
    20.1.1.1 from 20.1.1.1 (20.1.1.1)
      Origin IGP, localpref 100, valid, external, best
      unknown transitive attribute: flag 0xE0 type 0x62 length 0x64
        value 0000 0000 0100 0000 0200 0000 0300 0000
              0400 0000 0500 0000 0600 0000 0700 0000
              0800 0000 0900 0000 0A00 0000 0B00 0000
              0C00 0000 0D00 0000 0E00 0000 0F00 0000
              1000 0000 1100 0000 1200 0000 1300 0000
              1400 0000 1500 0000 1600 0000 1700 0000
              1800 0000
      rx pathid: 0, tx pathid: 0x0
      Updated on Jul 20 2019 07:50:43 PST
```

Tuning BGP

You can tune BGP characteristics through a series of optional parameters.

To tune BGP, use the following optional commands in router configuration mode:

Command	Purpose
bestpath [**always-compare-med** | **as-pathmultipath-relax** | **compare-routerid** |**cost-community ignore** | **igp-metric ignore** |**med** {**confed** |**missing-as-worst**| **non-deterministic**}] **Example:** `switch(config-router)# bestpath always-compare-med` **Note** If BGP computes an ECMP route, it is installed in Cisco Nexus 3550-T hardware as a unipath and a "Failure to install ECMP" warning sys log is generated.	Modifies the best-path algorithm. The optional parameters are as follows: • **always-compare-med** —Compares MED on paths from different autonomous systems. • **as-path multipath-relax** —Allows load sharing across the providers with different (but equal-length) AS paths. Without this option, the AS paths must be identical for load sharing. • **compare-routerid** —Compares the router IDs for identical eBGP paths. • **cost-community ignore** —Ignores the cost community for BGP best-path calculations. • **igp-metric ignore** —Ignores the Interior Gateway Protocol (IGP) metric for next hop during best-path selection. • **med confed** —Forces bestpath to do a MED comparison only between paths originated within a confederation. • **med missing-as-worst** —Treats a missing MED as the highest MED. • **med non-deterministic** —Does not always pick the best MED path from among the paths from the same autonomous system.
enforce-first-as **Example:** `switch(config-router)# enforce-first-as`	Enforces the neighbor autonomous system to be the first AS number listed in the AS_path attribute for eBGP.
log-neighbor-changes **Example:** `switch(config-router)# log-neighbor-changes`	Generates a system message when any neighbor changes state. **Note** To suppress neighbor status change messages for a specific neighbor, you can use the **log-neighbor-changes disable** command in router address-family configuration mode.
router-id *id* **Example:** `switch(config-router)# router-id 10.165.20.1`	Manually configures the router ID for this BGP speaker.

Command	Purpose
timers [**bestpath-delay** *delay* | *bgpkeepalive holdtime* | **prefix-peer-timeout** *timeout*] **Example:** `switch(config-router)# timers bgp 90 270`	Sets BGP timer values. The optional parameters are as follows: • *delay* —Initial best-path timeout value after a restart. The range is from 0 to 3600 seconds. The default value is 300. • *keepalive* —BGP session keepalive time. The range is from 0 to 3600 seconds. The default value is 60. • *holdtime* —BGP session hold time. The range is from 0 to 3600 seconds. The default value is180. • *timeout* —Prefix peer timeout value. The range is from 0 to 1200 seconds. The default value is 30. You must manually reset the BGP sessions after configuring this command.

To tune BGP, use the following optional commands in router address-family configuration mode:

Command	Purpose
distance *ebgp-distance ibgp-distance local-distance* **Example:** `switch(config-router-af)# distance 20 100 200`	Sets the administrative distance for BGP. The range is from 1 to 255. The defaults are as follows: • *ebgp-distance* —20. • *ibgp-distance* —200. • *local-distance* —220. Local-distance is the administrative distance used for aggregate discard routes when they are installed in the RIB. After you enter the value for the external administrative distance, you must enter the value for the administrative distance for the internal routes or/and the value for the administrative distance for the local routes depending on your requirement; so that the internal/local routes are also considered in the route administration.
log-neighbor-changes [**disable**] **Example:** `switch(config-router-af)# log-neighbor-changes disable`	Generates a system message when this specific neighbor changes state. The **disable** option suppresses neighbor status changes messages for this specific neighbor.

To tune BGP, use the following optional commands in neighbor configuration mode:

Command	Purpose
description *string* **Example:** `switch(config-router-neighbor)# description main site`	Sets a descriptive string for this BGP peer. The string can be up to 80 alphanumeric characters.
low-memory exempt **Example:** `switch(config-router-neighbor)# low-memory exempt`	Exempts this BGP neighbor from a possible shutdown due to a low memory condition.
transport connection-mode passive **Example:** `switch(config-router-neighbor)# transport connection-mode passive`	Allows a passive connection setup only. This BGP speaker does not initiate a TCP connection to a BGP peer. You must manually reset the BGP sessions after configuring this command.
[**no** | **default**] **remove-private-as** [**all** |**replace-as**] **Example:** `switch(config-router-neighbor)# remove-private-as`	Removes private AS numbers from outbound route updates to an eBGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions. The optional parameters are as follows: • **no** —Disables the command. • **default** —Moves the command to its default mode. • **all** —Removes all private-as numbers from the AS-path value. • **replace-as** —Replaces all private AS numbers with the replace-as AS-path value. See the Guidelines and Limitations for Advanced BGP, on page 293 section for additional information on this command.
update-source *interface-type number* **Example:** `switch(config-router-neighbor)# update-source ethernet 1/1`	Configures the BGP speaker to use the source IP address of the configured interface for BGP sessions to the peer. This command triggers an automatic notification and session reset for the BGP neighbor sessions. Single-hop iBGP peers support fast external fallover when **update-source** is configured.

To tune BGP, use the following optional commands in neighbor address-family configuration mode:

Command	Purpose
allowas in **Example:** `switch(config-router-neighbor-af)# allowas in`	Allows routes that have their own AS in the AS path to be installed in the BRIB.
default-originate [**route-map** *map-name*] **Example:** `switch(config-router-neighbor-af)# default-originate`	Generates a default route to the BGP peer.
disable-peer-as-check **Example:** `switch(config-router-neighbor-af)# disable-peer-as-check`	Disables peer AS-number checking while the device advertises routes learned from one node to another node in the same AS path.

filter-list *list-name* {**in** | **out**}
Example:
`switch(config-router-neighbor-af)# filter-list BGPFilter in`
	Applies an AS_path filter list to this BGP peer for inbound or outbound route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

prefix-list *list-name* {**in** | **out**}
Example:
`switch(config-router-neighbor-af)# prefix-list PrefixFilter in`
	Applies a prefix list to this BGP peer for inbound or outbound route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

send-community **Example:** `switch(config-router-neighbor-af)# send-community`	Sends the community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
send-community extended **Example:** `switch(config-router-neighbor-af)# send-community extended`	Sends the extended community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
suppress-inactive **Example:** `switch(config-router-neighbor-af)# suppress-inactive`	Advertises the best (active) routes only to the BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

Command	Purpose
[no | default] as-override **Example:** `switch(config-router-neighbor-af)# as-override`	**no** - (Optional) Disables the command. **default** - (Optional) Moves the command to its default mode. **as-override** - While sending updates to eBGP peer, replaces in the *path* attribute all occurrences of the peer's AS number with the local AS number.

Configuring Policy-Based Administrative Distance

You can configure a distance for external BGP (eBGP) and internal BGP (iBGP) routes that match a policy described in the configured route map. The distance configured in the route map is downloaded to the unicast RIB along with the matching routes. BGP uses the best path to determine the administrative distance when downloading next hops in the unicast RIB table. If there is no match or a deny clause in the policy, BGP uses the distance configured in the distance command or the default distance for routes.

The policy-based administrative distance feature is useful when there are two or more different routes to the same destination from two different routing protocols.

Before you begin

You must enable BGP.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip prefix-list** *name* **seq** *number* **permit** *prefix-length*	Creates a prefix list to match IP packets or routes with the permit keyword.
Step 3	switch(config)# **route-map** *map-tag* **permit** *sequence-number*	Creates a route map and enters route-map configuration mode with the permit keyword. If the match criteria for the route is met in the policy, the packet is policy routed.
Step 4	switch(config-route-map)# **match ip address prefix-list** *prefix-list-name*	Matches IPv4 network routes based on a prefix list. The prefix-list name can be any alphanumeric string up to 63 characters.
Step 5	switch(config-route-map)# **set distance** *value1 value2 value3*	Specifies the administrative distance for interior BGP (iBGP) or exterior BGP (eBGP) routes and BGP routes originated in the local autonomous system. The range is from 1 to 255. After you enter the value for the external administrative distance, you must enter the value for the administrative distance for the

	Command or Action	Purpose
		internal routes or/and the value for the administrative distance for the local routes depending on your requirement; so that the internal/local routes are also considered in the route administration.
Step 6	switch(config-route-map)# **exit**	Exits route-map configuration mode.
Step 7	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the AS number to the local BGP speaker.
Step 8	switch(config-router)# **address-family** {**ipv4** | **vpnv4**} **unicast**	Enters address family configuration mode.
Step 9	switch(config-router-af)# **table-map** *map-name*	Configures the selective administrative distance for a route map for BGP routes before forwarding them to the RIB table. The table-map name can be any alphanumeric string up to 63 characters. **Note** You can also configure the **table-map** command under the VRF address-family configuration mode.
Step 10	(Optional) switch(config-router-af)# **show forwarding distribution**	Displays forwarding information distribution.
Step 11	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configuring Multiprotocol BGP

You can configure MP-BGP to support multiple address families, including IPv4 unicast and multicast routes.

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		Enters BGP mode and assigns the autonomous system number to the local BGP speaker.

	Command or Action	Purpose

Step 3	**neighbor** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65534`
	`switch(config-router-neighbor)#`
		Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.

Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	Saves this configuration change.

Example

Configuring BMP

You can configure BMP on the Cisco Nexus® 3550-T device.

Before you begin

You must enable BGP (see the Enabling BGP section).

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp as-number**
	Example:
	`switch(config)# router bgp 200`
	`switch(config-router)#`
		Enters BGP mode and assigns the autonomous system number to the local BGP speaker.

Step 3	**bmp server** *server-number* **Example:** `switch(config-router)# bmp server 1`	Configures the BMP server to which BGP should send information. The server number is used as a key. **Note** You can configure up to two BMP servers.
Step 4	**address ip-address port-number port-number** **Example:**	Configures the IPv4 address of the host and the port number on which the BMP speaker connects to the BMP server.

	Command or Action	Purpose
	`switch(config-router)# address 10.1.1.1 port-number 2000`	
Step 5	**description** *string* **Example:** `switch(config-router)# description BMPserver1`	Configures the BMP server description. You can enter up to 256 alphanumeric characters.

Step 6	**initial-refresh** { *skip* | *delay time*}
	Example:
	`switch(config-router)# initial-refresh delay 100`
		Configures the option to send a route refresh when BGP is converged and the BMP server connection is established later.
		The skip option specifies to not send a route refresh if the BMP server connection comes up later.
		The delay option specifies the time in seconds after which the route refresh should be sent. The range is from 30 to 720 seconds, and the default value is 30 seconds.

Step 7	**initial-delay** *time* **Example:** `switch(config-router)# initial-delay 120`	Configures the delay after which a connection is attempted to the BMP server. The range is from 30 to 720 seconds, and the default value is 45 seconds.
Step 8	**stats-reporting-period** *time* **Example:** `switch(config-router)# stats-reporting-period 50`	Configures the time interval in which the BMP server receives the statistics report from BGP neighbors. The range is from 30 to 720 seconds, and the default is disabled.
Step 9	shutdown **Example:** `switch(config-router)# shutdown`	Disables the connection to the BMP server.

Step 10	**neighbor ip-address**
	Example:
	`switch(config-router)# neighbor 192.168.1.2`
	`switch(config-router-neighbor)#`
		Enters neighbor configuration mode for BGP routing and configures the neighbor IP address.

Step 11	**remote-as** *as-number* **Example:** `switch(config-router-neighbor)# remote-as 65535`	Configures the AS number for a remote BGP peer.
Step 12	**bmp-activate-server** *server-number* **Example:** `switch(config-router-neighbor)# bmp-activate-server 1`	Configures the BMP server to which a neighbor's information should be sent.

	Command or Action	Purpose
Step 13	(Optional) **show bgp bmp** *server [server-number] [detail]* **Example:** `switch(config-router-neighbor)# show bgp bmp server`	Displays BMP server information.
Step 14	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor)# copy running-config startup-config`	Saves this configuration change.

About BGP Graceful Shutdown

BGP supports the graceful shutdown feature. This BGP feature works with the BGP **shutdown** command to:

- Dramatically decrease the network convergence time when a router or link is taken offline.
- Reduce or eliminate dropped packets that are in transit when a router or link is taken offline.

Despite the name, BGP graceful shutdown does not actually cause a shutdown. Instead, it alerts connected routers that a router or link will be going down soon.

The graceful shutdown feature uses the GRACEFUL_SHUTDOWN well-known community (0xFFFF0000 or 65535:0), which is identified by IANA and the IETF through RFC 8326. This well-known community can be attached to any routes, and it is processed like any other attribute of a route.

Because this feature announces that a router or link will be going down, the feature is useful in preparation of maintenance windows or planned outages. Use this feature before shutting down BGP to limit the impact on traffic.

Graceful Shutdown Aware and Activate

BGP routers can control the preference of all routes with the GRACEFUL_SHUTDOWN community through the concept of GRACEFUL SHUTDOWN awareness. Graceful shutdown awareness is enabled by default, which enables the receiving peers to deprefer incoming routes carrying the GRACEFUL_SHUTDOWN community. Although not a typical use case, you can disable and reenable graceful shutdown awareness through the **graceful-shutdown aware** command.

Graceful shutdown aware is applicable only at the BGP global context. For information about contexts, see Graceful Shutdown Contexts, on page 337. The aware option operates with another option, the **activate** option, which you can assign to a route map for more granular control over graceful shutdown routes.

Interaction of the Graceful Shutdown Aware and Activate Options

When a graceful shutdown is activated, the GRACEFUL_SHUTDOWN community is appended to route updates only when you specify the **activate** keyword. At this point, new route updates that contain the community are generated and transmitted. When the **graceful-shutdown aware** command is configured, all routers that receive the community then deprefer (lower the route preference of) the routes in the update. Without the **graceful-shutdown aware** command, BGP does not deprefer routes with the GRACEFUL_SHUTDOWN community.

After the feature is activated and the routers are aware of graceful shutdown, BGP still considers the routes with the GRACEFUL_SHUTDOWN community as valid. However, those routes are given the lowest priority in the best-path calculation. If alternate paths are available, new best paths are chosen, and convergence occurs to accommodate the router or link that will soon go down.

Graceful Shutdown Contexts

BGP graceful shutdown feature has two contexts that determine what the feature affects and what functionality is available.

Context	Affects	Commands
Global	The entire switch and all routes processed by it. For example, readvertise all routes with the GRACEFUL_SHUTDOWN community.	**graceful-shutdown activate [route-map** *route-map***]** **graceful-shutdown aware**
Peer	A BGP peer or a link between neighbors. For example, advertise only one link between peers with GRACEFUL_SHUTDOWN community.	**graceful-shutdown activate [route-map** *route-map***]**

Graceful Shutdown with Route Maps

Graceful shutdown works with the route policy manager (RPM) feature to control how the switch's BGP router transmits and receives routes with the GRACEFUL_SHUTDOWN community. Route maps can process route updates with the community in the inbound and outbound directions. Typically, route maps are not required. However, if needed, you can use them to customize the control of graceful shutdown routes.

Normal Inbound Route Maps

Normal inbound route maps affect routes that are incoming to the BGP router. Normal inbound route maps are not commonly used with the graceful shutdown feature because routers are aware of graceful shutdown by default.

Cisco Nexus® switches do not require an inbound route map for the graceful shutdown feature. Cisco NX-OS switch have implicit inbound route maps that automatically deprefer any routes that have the GRACEFUL_SHUTDOWN community if the BGP router is graceful shutdown aware.

Normal inbound route maps can be configured to match against the well-known GRACEFUL_SHUTDOWN community. Although these inbound route maps are not common, there are some cases where they are used:

- If switches are running a Cisco NX-OS release that do not have the implicit inbound route map, a graceful shutdown inbound route map to use the graceful shutdown feature on these switches. The route map must match inbound routes with the well-known GRACEFUL_SHUTDOWN community, permit them, and deprefer them. If an inbound route map is needed, create it on the BGP peer that is running a compatible version of NX-OS and is receiving the graceful shutdown routes.

- If you want to disable graceful shutdown aware, but still want the router to act on incoming routes with GRACEFUL_SHUTDOWN community from some BGP neighbors, you can configure an inbound route map under the respective peers.

Normal Outbound Route Maps

Normal outbound route maps control forwarding the routes that a BGP router sends. Normal outbound route maps can affect the graceful shutdown feature. For example, you can configure an outbound route map to match on the GRACEFUL_SHUTDOWN community and set attributes, and it takes precedence over any graceful shutdown outbound route maps.

Graceful Shutdown Outbound Route Maps

Outbound Graceful shutdown route maps are specific type of outbound route map for the graceful shutdown feature. They are optional, but they are useful when you already have a community list that is associated with a route map. The typical graceful shutdown outbound route map contains only `set` clauses to set or modify certain attributes.

You can use outbound route maps in the following ways:

- For customers that already have existing outbound route maps, you can add a new entry with a higher sequence number, match on the GRACEFUL_SHUTDOWN well-known community, and add any attributes that you want.
- You can also use a graceful shutdown outbound route map with the **graceful-shutdown activate route-map** *name* option. This is the typical use case.

  This route map requires no match clauses, so the route map matches on all routes being sent to the neighbor.

Route Map Precedence

When multiple route maps are present on the same router, the following order of precedence is applied to determine how routes with the community are processed:Consider the following example. Assume you have a standard outbound route map name Red that sets a local-preference of 60. Also, assume you have a peer graceful-shutdown route map that is named Blue that sets local-pref to 30. When the route update is processed, the local preference will be set to 60 because Red overwrites Blue.

- Normal outbound route maps take precedence over peer graceful shutdown maps.
- Peer graceful shutdown maps take precedence over global graceful shutdown maps.

Guidelines and Limitations

The following are limitations and guidelines for BGP global shutdown:

- Graceful shutdown feature can only help avoid traffic loss when alternative routes exist in the network for the affected routers. If the router has no alternate routes, routes carrying the GRACEFUL_SHUTDOWN community are the only ones available, and therefore, are used in the best-path calculation. This situation defeats the purpose of the feature.

- Configuring a BGP send community is required to send the GRACEFUL_SHUTDOWN community.
- For route maps:
    - When global route maps and neighbor route maps are configured, the per-neighbor route maps take precedence.
    - Outbound route maps take precedence over any global route maps configured for graceful shutdown.
    - Outbound route maps take precedence over any peer route maps configured for graceful shutdown.
    - To add the graceful shutdown functionality to legacy (existing) inbound route maps, follow this order:
        1. Add the graceful shutdown match clause to the top of the route map by setting a low sequence number for the clause (for example, sequence number 0).
        2. Add a continue statement after the graceful shutdown clause. If you omit the continue statement, route-map processing stops when it matches the graceful shutdown clause, any other clauses with higher sequence numbers (for example, 1 and higher) are not processed.

Graceful Shutdown Task Overview

To use the graceful shutdown feature, you typically enable graceful-shutdown aware on all Cisco Nexus switches and leave the feature enabled. When a BGP router must be taken offline, you configure graceful-shutdown activate on it.

The following details document the best practice for using the graceful shutdown feature.

To bring the router or link down:

1. Configure the Graceful Shutdown feature.
2. Watch the neighbor for the best path.
3. When the best path is recalculated, issue the **shutdown** command to disable BGP.
4. Perform the work that required you to shut down the router or link.

To bring the router or link back online:

1. When you finish the work that required the shutdown, reenable BGP (**no shutdown**).
2. Disable the graceful shutdown feature (**no graceful-shutdown activate** in config router mode).

Configuring Graceful Shutdown on a Link

This task enables you to configure graceful shutdown on a specific link between two BGP routers.

Before you begin

If you have not already enabled BGP, enable it now (**feature bgp**).

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** *autonomous-system-number*
	Example:
	`switch-1(config)#` **`router bgp 110`**
	`switch-1(config-router)#`
		Enters router configuration mode to create or configure a BGP routing process.

Step 3	**neighbor {** *ipv4-address*} **remote-as** *as-number*
	Example:
	`switch-1(config-router)#` **`neighbor 10.0.0.3 remote-as 200`**
	`switch-1(config-router-neighbor)#`
		Configures the autonomous system (AS) to which the neighbor belongs.

Step 4	**graceful-shutdown activate [route-map** *map-name***]**
	Example:
	`switch-1(config-router-neighbor)#` **`graceful-shutdown activate route-map gshutPeer out`**
	`switch-1(config-router-neighbor)#`
		Configures graceful shutdown on the link to the neighbor. Also, advertises the routes with the well-known GRACEFUL_SHUTDOWN community and applies the route map to the outbound route updates.
		The routes are advertised with the graceful-shutdown community by default. In this example, routes are advertised to the neighbor with the Graceful-shutdown community with a route-map named gshutPeer.
		The devices receiving the gshut community look at the communities of the route and optionally use the communities to apply routing policy.

Filtering BGP Routes and Setting Local Preference Based On GRACEFUL_SHUTDOWN Communities

Switches do not have an inbound route map that matches against the GRACEFUL_SHUTDOWN community name. Therefore, they have no way of identifying and depreferring the correct routes.

For switches running a release of NX-OS, you must configure an inbound route map that matches on the community value for graceful shutdown (65535:0) and deprefers routes.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch-1# ` **`configure terminal`**
	`switch-1<config)#`
		Enters global configuration mode.

Step 2	**ip community list standard** *community-list-name* **seq** *sequence-number* **{ permit | deny }** *value*
	Example:
	`switch-1(config)# ` **`ip community-list standard GSHUT seq 10 permit 65535:0`**
	`switch-1(config)#`
		Configures a community list and permits or denies routes that have the well-known graceful shutdown community value.

Step 3	**route map** *map-tag* **{deny | permit}** *sequence-number*
	Example:
	`switch-1(config)# ` **`route-map RM_GSHUT permit 10`**
	`switch-1(config-route-map)#`
		Configures a route map as sequence 10 and permits routes that have the GRACEFUL_SHUTDOWN community.

Step 4	**match community** *community-list-name*
	Example:
	`switch-1(config-route-map)# ` **`match community GSHUT`**
	`switch-1(config-route-map)#`
		Configures that routes that match the IP community list GSHUT are processed by Route Policy Manager (RPM).

Step 5	**set local-preference** *local-pref-value*
	Example:
	`switch-1(config-route-map)# ` **`set local-preference 10`**
	`switch-1(config-route-map)#`
		Configures that the routes that match the IP community list GSHUT will be given a specified local preference.

Step 6	**exit**
	Example:
	`switch-1(config-route-map)# ` **`exit`**
	`switch-1(config)#`
		Leaves route map configuration and returns to global configuration mode.

Step 7	**router bgp** *community-list-name*
	Example:
	`switch-1(config)# ` **`router bgp 100`**
	`switch-1(config-router)#`
		Enters router configuration mode and creates a BGP instance.

Step 8	**neighbor {** *ipv4-address* **}**
	Example:
	`switch-1(config-router)# ` **`neighbor 10.0.0.3`**
	`switch-1(config-router-neighbor)#`
		Enters route BGP neighbor mode for a specified neighbor.

	Command or Action	Purpose

Step 9	**address-family {** *address-family sub family* **}**
	Example:
	`nxosv2(config-router-neighbor)#`
	`address-family ipv4 unicast`
	`nxosv2(config-router-neighbor-af)#`
		Puts the neighbor into address family (AF) configuration mode.

Step 10	**send community**
	Example:
	`nxosv2(config-router-neighbor-af)#`
	`send-community`
	`nxosv2(config-router-neighbor-af)#`
		Enables BGP community exchange with the neighbor.

Step 11	**route map** *map-tag* **in**
	Example:
	`nxosv2(config-router-neighbor-af)#`
	`route-map RM_GSHUT in`
	`nxosv2(config-router-neighbor-af)#`
		Applies the route map to incoming routes from the neighbor. In this example, the route map that is named RM_GSHUT permits routes with the GRACEFUL_SHUTDOWN community from the neighbor.

Configuring Graceful Shutdown for All BGP Neighbors

You can manually apply the GRACEFUL_SHUTDOWN well-known community to all the neighbors of a graceful shutdown initiator.

You can configure graceful shutdown at the global level for all BGP neighbors.

Before you begin

If you have not already enabled BGP, enable it now (**feature bgp**).

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** *autonomous-system-number*
	Example:
	`switch-1(config)#` **`router bgp 110`**
	`switch-1(config-router)#`
		Enters router configuration mode to create or configure a BGP routing process.

Step 3	**graceful-shutdown activate [route-map** *map-name***]** **Example:**	Configures graceful shutdown route map for the links to all neighbors. Also, advertises all routes with the well-known GRACEFUL_SHUTDOWN community and

	Command or Action	Purpose
	switch-1(config-router-neighbor)# **graceful-shutdown activate route-map gshutPeer** switch-1(config-router-neighbor)#	applies the route map to the outbound route updates. The routes are advertised with the GRACEFUL_SHUTDOWN community by default. In this example, routes are advertised to all neighbors with the community with a route-map named gshutPeer. The route map should contain only set clauses. The devices receiving the GRACEFUL_SHUTDOWN community look at the communities of the route and optionally use the communities to apply routing policy.

Controlling the Preference for All Routes with the GRACEFUL_SHUTDOWN Community

Cisco NX-OS enables lowering the preference of incoming routes that have the GRACEFUL_SHUTDOWN community. When **graceful shutdown aware** is enabled, BGP considers routes carrying the community as the lowest preference during best path calculation. By default, lowering the preference is enabled, but you can selectively disable this option.

Whenever you enable or disable this option, you trigger a BGP best-path calculation. This option gives you the flexibility to control the behavior of the BGP best-path calculation for the graceful shutdown well-known community.

Before you begin

If you have not enabled BGP, enable it now (**feature bgp**).

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch-1(config)# **config terminal** switch-1(config)#	Enters global configuration mode.
Step 2	**router bgp** *autonoums-system* **Example:** switch-1(config)# **router bgp 100** switch-1(config-router)#	Enters router configuration mode and configures a BGP routing process.
Step 3	(Optional) **no graceful-shutdown aware** **Example:**	For this BGP router, do not give lower preference for all routes that have the GRACEFUL_SHUTDOWN community. The

	Command or Action	Purpose

	`switch-1(config-router)# no graceful-shutdown aware`
	`switch-1(config-router)#`
		default action is to deprefer routes when the graceful shutdown aware feature is disabled, so using the **no** form of the command is optional for not depreferring graceful shutdown routes.

Preventing Sending the GRACEFUL_SHUTDOWN Community to a Peer

If you no longer need the GRACEFUL_SHUTDOWN community that is appended as a route attribute to outbound route updates, you can remove the community, which no longer sends it to a specified neighbor. One use case would be when a router is at an autonomous system boundary, and you do not want the graceful shutdown functionality to propagate outside of an autonomous system boundary.

To prevent sending the GRACEFUL_SHUTDOWN to a peer, you can disable the send community option or strip the community from the outbound route map.

Choose either of the following methods:

- Disable the send-community in the running config.

  Example:

  ```
  nxosv2(config-router-neighbor-af)# no send-community standard
  nxosv2(config-router-neighbor-af)#
  ```

  If you use this option, the GRACEFUL_SHUTDOWN community is still received by the switch, but it is not sent to the downstream neighbor through the outbound route map. All standard communities are not sent either.

- Delete the GRACEFUL_SHUTDOWN community through an outbound route map by following these steps:

  1. Create an IP community list matches the GRACEFUL_SHUTDOWN community.
  2. Create an outbound route map to match against the GRACEFUL_SHUTDOWN community.
  3. Use a **set community-list delete** clause to strip GRACEFUL_SHUTDOWN community.

  If you use this option, the community list matches and permits the GRACEFUL_SHUTDOWN community, then the outbound route map matches against the community and then deletes it from the outbound route map. All other communities pass through the outbound route map without issue.

Displaying Graceful Shutdown Information

Information about the graceful shutdown feature is available through the following **show** commands.

Command	Action
show ip bgp community-list graceful-shutdown	Shows all entires in the BGP routing table that have the GRACEFUL_SHUTDOWN community.

Command	Action
show running-config bgp	Shows the running BGP configuration.
show running-config bgp all	Shows all information for the running BGP configuration including information about the graceful shutdown feature.
show bgp *address-family* **neighbors** *neighbor-address* **Note** In Cisco Nexus 3550-T BGP only supports IPv4 unicast address family.	When the feature is configured for the peer, shows the following: • The state of the graceful-shutdown-activate feature for the specified neighbor • The name of any graceful shutdown route map configured for the specified neighbor
show bgp process	Shows different information depending on the context. When the graceful-shutdown-activate option is configured in peer context, shows the enabled or disabled state for the feature through `graceful-shutdown-active`. When the graceful-shutdown-activate option is configured in global context and has a graceful-shutdown route map, shows the enabled state of the feature through the following: • `graceful-shutdown-active` • `graceful-shutdown-aware` • `graceful-shutdown route-map`
show ip bgp *address*	For the specified address, shows the BGP routing table information, including the following: • The state of the specified address as the best path • Whether the specified address is part of the GRACEFUL_SHUTDOWN community

Graceful Shutdown Configuration Examples

These examples show some configurations for using the graceful shutdown feature.

Configuring Graceful Shutdown for a BGP Link

The following example shows how to configure graceful shutdown while setting a local preference and a community:

- Configuring graceful shutdown activate for the link to the specified neighbor

- Adding the GRACEFUL_SHUTDOWN community to the routes
- Setting a route map named gshutPeer with only set clauses for outbound routes with the community.

```
router bgp 100
    neighbor 20.0.0.3 remote-as 200
         graceful-shutdown activate route-map gshutPeer
         address-family ipv4 unicast
              send-community

route-map gshutPeer permit 10
    set local-preference 0
    set community 200:30
```

Configuring Graceful Shutdown for All-Neighbor BGP Links

The following example shows:

- Configuring graceful shutdown activate for all the links connecting the local router and all its neighbors.
- Adding the GRACEFUL_SHUTDOWN community to the routes.
- Setting a route map that is named gshutAall with only set clauses for all outbound routes.

```
router bgp 200
   graceful-shutdown activate route-map gshutAll

route-map gshutAll permit 10
   set as-path prepend 10 100 110
   set community 100:80

route-map Red permit 10
   set local-pref 20

router bgp 100
   graceful-shutdown activate route-map gshutAll
      router-id 2.2.2.2
         address-family ipv4 unicast
         network 2.2.2.2/32
         neighbor 1.1.1.1 remote-as 100
         update-source loopback0
         address-family ipv4 unicast
            send-community
         neighbor 20.0.0.3 remote-as 200
         address-family ipv4 unicast
            send-community
               route-map Red out
```

In this example, the `gshutAll` route-map takes effect for neighbor 1.1.1.1, but not neighbor 20.0.0.3, because the outbound route-map `Red` configured under neighbor 20.0.0.3 takes precedence instead.

Configuring Graceful Shutdown Under a Peer-Template

This example configures the graceful shutdown feature under a peer-session template, which is inherited by a neighbor.

```
router bgp 200
   template peer-session p1
      graceful-shutdown activate route-map gshut_out
   neighbor 1.1.1.1 remote-as 100
      inherit peer-session p1
      address-family ipv4 unicast
         send-community
```

Filtering BGP Routes and Setting Local Preference Based on GRACEFUL_SHUTDOWN Community Using and Inbound Route Map

This example shows how to use a community list to filter the incoming routes that have the GRACEFUL_SHUTDOWN community. This configuration is useful for legacy switches that are not running Cisco NX-OS 9.3(1) as a minimum version.

The following example shows:

- An IP Community List that permits routes that have the GRACEFUL_SHUTDOWN community.
- A route map that is named RM_GSHUT that permits routes based on a standard community list named GSHUT.
- The route map also sets the preference for the routes it processes to 0 so that those routes are given lower preference for best path calculation when the router goes offline. The route map is applied to incoming IPv4 routes from the neighbor (20.0.0.2).

```
ip community-list standard GSHUT permit 65535:0

route-map RM_GSHUT permit 10
   match community GSHUT
   set local-preference 0

router bgp 200
   neighbor 20.0.0.2 remote-as 100
      address-family ipv4 unicast
         send-community
         route-map RM_GSHUT in
```

Configuring a Graceful Restart

You can configure a graceful restart and enable the graceful restart helper feature for BGP.

Before you begin

You must enable BGP (see the Enabling BGP section).

Create the VRFs.

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** *as-number* **Example:**	Creates a new BGP process with the configured autonomous system number.

	Command or Action	Purpose

	`switch(config)# router bgp 65535`
	`switch(config-router)#`	

Step 3	(Optional) **timers prefix-peer-timeout** *timeout* **Example:** `switch(config-router)# timers prefix-peer-timeout 20`	Configures the timeout value (in seconds) for BGP prefix peers. The default value is 90 seconds.
Step 4	**graceful-restart** **Example:** `switch(config-router)# graceful-restart`	Enables a graceful restart and the graceful restart helper functionality. This command is enabled by default. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

Step 5	**graceful-restart** {**restart-time** *time*|**stalepath-time** *time*}
	Example:
	`switch(config-router)# graceful-restart restart-time 300`
		Configures the graceful restart timers.
		The optional parameters are as follows:
		• **restart-time**—Maximum time for a restart sent to the BGP peer. The range is from 1 to 3600 seconds. The default is 120.
		• **stalepath-time**—Maximum time that BGP keeps the stale routes from the restarting BGP peer. The range is from 1 to 3600 seconds. The default is 300.
		This command triggers an automatic notification and session reset for the BGP neighbor sessions.

Step 6	**graceful-restart-helper** **Example:** `switch(config-router)# graceful-restart restart-time 300`	Enables the graceful restart helper functionality. Use this command if you have disabled graceful restart but you still want to enable graceful restart helper functionality. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 7	(Optional) **show running-config bgp** **Example:** `switch(config-router)# show running-config bgp`	Displays the BGP configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-router)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to enable a graceful restart:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# graceful-restart
switch(config-router)# graceful-restart restart-time 300
switch(config-router)# copy running-config startup-config
```

Verifying the Advanced BGP Configuration

To display the BGP configuration, perform one of the following tasks:

Note ***Cisco Nexus 3550-T - 10.1(2t) release***, supports only default VRF and management VRF.

Command	Purpose
show bgp all [**summary**]	Displays the BGP information for all address families.
show bgp convergence	Displays the BGP information for all address families.
show bgp {**ipv4**} {**unicast**} [*ip-address*] **community** {**regexp** *expression* | [**community**] [**no-advertise**] [**no-export**] [**no-export-subconfed**]}	Displays the BGP routes that match a BGP community.
show bgp{**ipv4**} {**unicast**} [*ip-address*] **community-list** *list-name*	Displays the BGP routes that match a BGP community list.
show bgp {**ipv4**} {**unicast**} [*ip-address*] **extcommunity** {**regexp** *expression* | **generic** [**non-transitive** | **transitive**] *aa4:nn* [**exact-match**]}	Displays the BGP routes that match a BGP extended community.
show bgp {**ipv4**} {**unicast**} [*ip-address*] **extcommunity-list** *list-name* [**exact-match**]}	Displays the BGP routes that match a BGP extended community list.
show bgp {**ipv4**} {**unicast**} [*ip-address*] **extcommunity-list** *list-name* [**exact-match**]}	Displays the information for BGP route dampening. Use the **clear bgp dampening** command to clear the route flap dampening information.
show bgp {**ipv4**} {**unicast**} [*ip-address*] {**dampening dampened-paths** [**regexp** *expression*]}	Displays the BGP route history paths.
show bgp {**ipv4** | **vpnv4**} {**unicast**} [*ip-address*] **filter-list** *list-name*	Displays the information for the BGP filter list.

Command	Purpose
show bgp {**ipv4** | **vpnv4**} {**unicast**} [*ip-address*] **neighbors** [*ip-address*]	Displays the information for BGP peers. Use the **clear bgp neighbors** command to clear these neighbors.
show bgp {**ipv4**} {**unicast**} [*ip-address*] {**nexthop** | **nexthop-database**}	Displays the information for the BGP route next hop.
show bgp paths	Displays the BGP path information.
show bgp {**ipv4**} {**unicast**} [*ip-address*] **policy** *name*	Displays the BGP policy information. Use the **clear bgp policy** command to clear the policy information.
show bgp {**ipv4**} {**unicast**} [*ip-address*] **prefix-list** *list-name*	Displays the BGP routes that match the prefix list.
show bgp {**ipv4**} {**unicast**} [*ip-address*] **received-paths**	Displays the BGP paths stored for soft reconfiguration.
show bgp {**ipv4** } {**unicast**} [*ip-address*] **regexp** *expression*	Displays the BGP routes that match the AS_path regular expression.
show bgp {**ipv4**} {**unicast**} [*ip-address*] **route-map** *map-name*	Displays the BGP routes that match the route map.
show bgp peer-policy *name*	Displays the information about BGP peer policies.
show bgp peer-session *name*	Displays the information about BGP peer sessions.
show bgp peer-template *name*	Displays the information about BGP peer templates. Use the **clear bgp peer-template** command to clear all neighbors in a peer template.
show bgp process	Displays the BGP process information.
show ip route *ip-address* **detail vrf all | i bw**	Displays the link bandwidth EXTCOMM fields. bw:xx (such as bw:40) in the output indicates that BGP peers are sending BGP extended attributes with the bandwidth.
show {**ipv4**} **bgp** *options*	Displays the BGP status and configuration information.

Command	Purpose
show running-configuration bgp	Displays the current running BGP configuration.

Monitoring BGP Statistics

To display BGP statistics, use the following commands:

Command	Purpose
	Displays the BGP route flap statistics. Use the **clear bgp flap-statistics** command to clear these statistics.
	Displays injected routes in the routing table.
show bgp sessions	Displays the BGP sessions for all peers. Use the **clear bgp sessions** command to clear these statistics.
show bgp statistics	Displays the BGP statistics.

Related Topics

The following topics can give more information on BGP:

- Configuring Basic BGP, on page 267

Additional References

For additional information related to implementing BGP, see the following sections:

CHAPTER 22

Configuring Static Routing

This chapter describes how to configure static routing on the Cisco NX-OS device.

This chapter contains the following sections:

About Static Routing

Routers forward packets using either route information from route table entries that you manually configure or the route information that is calculated using dynamic routing algorithms.

Static routes, which define explicit paths between two routers, cannot be automatically updated. You must manually reconfigure static routes when network changes occur. Static routes use less bandwidth than dynamic routes. No CPU cycles are used to calculate and analyze routing updates.

You can supplement dynamic routes with static routes where appropriate. You can redistribute static routes into dynamic routing algorithms, but you cannot redistribute routing information calculated by dynamic routing algorithms into the static routing table.

You should use static routes in environments where network traffic is predictable and where the network design is simple. You should not use static routes in large, constantly changing networks because static routes cannot react to network changes. Most networks use dynamic routes to communicate between routers but might have one or two static routes configured for special cases. Static routes are also useful for specifying a gateway of last resort (a default router to which all unroutable packets are sent).

Administrative Distance

An administrative distance is the metric used by routers to choose the best path when there are two or more routes to the same destination from two different routing protocols. An administrative distance guides the selection of one routing protocol (or static route) over another, when more than one protocol adds the same route to the unicast routing table. Each routing protocol is prioritized in order of most to least reliable using an administrative distance value.

Static routes have a default administrative distance of 1. A router prefers a static route to a dynamic route because the router considers a route with a low number to be the shortest. If you want a dynamic route to override a static route, you can specify an administrative distance for the static route. For example, if you have two dynamic routes with an administrative distance of 120, you would specify an administrative distance that is greater than 120 for the static route if you want the dynamic route to override the static route.

Directly Connected Static Routes

You must specify only the output interface (the interface on which all packets are sent to the destination network) in a directly connected static route. The router assumes that the destination is directly attached to the output interface and the packet destination is used as the next-hop address. The next hop can be an interface, but only for point-to-point interfaces. For broadcast interfaces, the next hop must be an IPv4 address.

Fully Specified Static Routes

You must specify either the output interface (the interface on which all packets are sent to the destination network) or the next-hop address in a fully specified static route. You can use a fully specified static route when the output interface is a multi-access interface and you need to identify the next-hop address. The next-hop address must be directly attached to the specified output interface.

Floating Static Routes

A floating static route is a static route that the router uses to back up a dynamic route. You must configure a floating static route with a higher administrative distance than the dynamic route that it backs up. In this instance, the router prefers a dynamic route to a floating static route. You can use a floating static route as a replacement if the dynamic route is lost.

Note By default, a router prefers a static route to a dynamic route because a static route has a smaller administrative distance than a dynamic route.

Remote Next Hops for Static Routes

You can specify the next-hop address of a neighboring router that is not directly connected to the router for static routes with remote (non-directly attached) next hops. If a static route has remote next hops during data forwarding, the next hops are recursively used in the unicast routing table to identify the corresponding directly attached next hops that have reachability to the remote next hops.

Prerequisites for Static Routing

Static routing has the following prerequisites:

- If the next-hop address for a static route is unreachable, the static route is not added to the unicast routing table.

Default Settings

The table lists the default settings for static routing parameters.

Table 20: Default Static Routing Parameters

Parameters	Default
Administrative distance	1
RIP feature	Disabled

Configuring Static Routing

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Configuring a Static Route

You can configure a static route on the device.

Procedure

	Command or Action	Purpose

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

Step 2	Enter the following command: **Example:** `switch(config)# ip route 192.0.2.0/8 ethernet 1/2 192.0.2.4`	**ip route** {*ip-prefix* | *ip-addr*/*ip-mask*} {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*preference*] Configures a static route and the interface for this static route. Use **?** to display a list of supported interfaces. You can specify a null interface by using **null 0**. You can optionally configure the next-hop address. The *preference* value sets the administrative distance. The range is from 1 to 255. The default is 1.

	Command or Action	Purpose
		Note Use the **no** {**ip**} **route** command to remove the static route.
Step 3	(Optional) **show** {**ip**} **static-route** **Example:** `switch(config)# show ip static-route`	Displays information about static routes.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to configure a static route for a null interface:

```
switch# configure terminal
switch(config)# ip route 1.1.1.1/32 null 0
switch(config)# copy running-config startup-config
```

Configuring a Static Route Over a VLAN

You can configure a static route without next-hop support over a VLAN.

Before you begin

Ensure that the access port is part of the VLAN.

Procedure

	Command or Action	Purpose

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

Step 2	**feature interface vlan** **Example:** `switch(config)# feature interface-vlan`	Enables VLAN interface mode.
Step 3	**interface-vlan** *vlan-id* **Example:** `switch(config)# interface-vlan 10`	Creates an SVI and enters interface configuration mode. The range for the **vlan-id** argument is from 1 to 4094, except for the VLANs reserved for the internal switch.

	Command or Action	Purpose
Step 4	**ip address** *ip-addr*/*length* **Example:** `switch(config)# ip address 192.0.2.1/8`	Configures an IP address for the VLAN.
Step 5	[**no**] **ip route** *ip-addr*/*length vlan-id* **Example:** `switch(config)# ip route 209.165.200.224/27 vlan 10`	Adds an interface static route without a next hop on the switch virtual interface (SVI). The IP address is the address that is configured on the interface that is connected to the switch. Use the **no** keyword with this command to remove the static route.
Step 6	(Optional) **show ip route** **Example:** `switch(config)# show ip route`	Displays routes from the Unicast Route Information Base (URIB).
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to configure a static route without a next hop over an SVI:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# interface vlan 10
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# ip route 209.165.200.224/27 vlan 10  <===209,165.200.224 is the IP
address of the interface that is configured on the interface that is directly connected to
the switch.
switch(config-if)# copy running-config startup-config
```

Verifying the Static Routing Configuration

To display the static routing configuration, perform one of the following tasks:

Command	Purpose
show {**ip**} **static-route**	Displays the configured static routes.
show {**ip**} **static-route track-table**	Displays information about the IPv4 static-route track table.

Configuration Example for Static Routing

This example shows how to configure static routing:

```
configure terminal
ip route 192.0.2.0/8 192.0.2.10
copy running-config startup-config
```

CHAPTER 23

Configuring VRRP

This chapter contains the following sections:

About VRRP

VRRP allows for a transparent failover at the first-hop IP router by configuring a group of routers to share a virtual IP address. VRRP selects an allowed router in that group to handle all packets for the virtual IP address. The remaining routers are in standby and take over if the allowed router fails.

VRRP Operation

A LAN client can determine which router should be the first hop to a particular remote destination by using a dynamic process or static configuration. Examples of dynamic router discovery are as follows:

Proxy ARP—The client uses Address Resolution Protocol (ARP) to get the destination it wants to reach, and a router responds to the ARP request with its own MAC address.

Routing protocol—The client listens to dynamic routing protocol updates (for example, from Routing Information Protocol [RIP]) and forms its own routing table.

ICMP Router Discovery Protocol (IRDP) client—The client runs an Internet Control Message Protocol (ICMP) router discovery client.

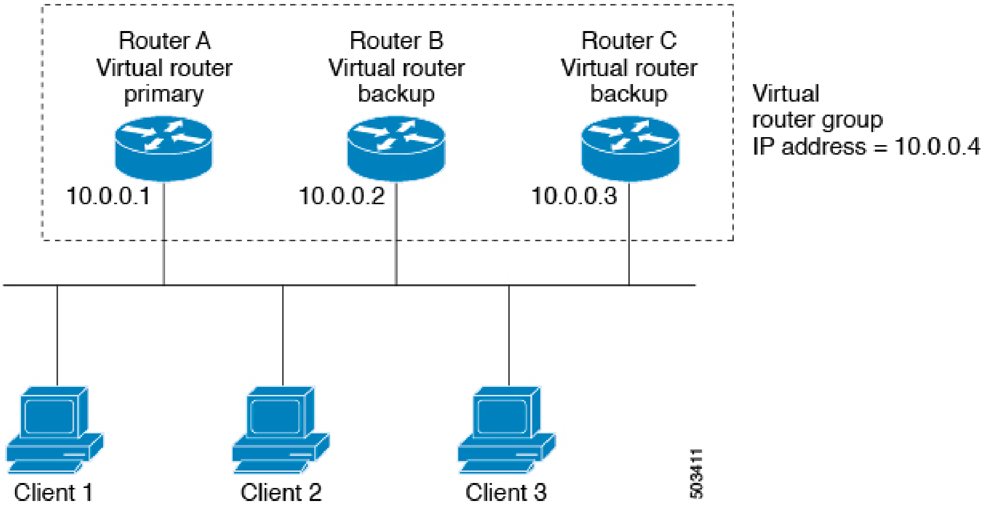
The disadvantage to dynamic discovery protocols is that they incur some configuration and processing overhead on the LAN client. Also, if a router fails, the process of switching to another router can be slow.

An alternative to dynamic discovery protocols is to statically configure a default router on the client. Although this approach simplifies client configuration and processing, it creates a single point of failure. If the default gateway fails, the LAN client is limited to communicating only on the local IP network segment and is cut off from the rest of the network.

VRRP can solve the static configuration problem by enabling a group of routers (a VRRP group) to share a single virtual IP address. You can then configure the LAN clients with the virtual IP address as their default gateway.

The following figure shows a basic VLAN topology. In this example, Routers A, B, and C form a VRRP group. The IP address of the group is the same address that was configured for the Ethernet interface of Router A (10.0.0.1).

Figure 20: Basic VRRP Topology



Because the virtual IP address uses the IP address of the physical Ethernet interface of Router A, Router A is the primary (also known as the IP address owner). As the primary, Router A owns the virtual IP address of the VRRP group and forwards packets sent to this IP address. Clients 1 through 3 are configured with the default gateway IP address of 10.0.0.1.

Routers B and C function as backups. If the primary fails, the backup router with the highest priority becomes the primary and takes over the virtual IP address to provide uninterrupted service for the LAN hosts. When Router A recovers, it becomes the primary again.

Note Packets received on a routed port destined for the VRRP virtual IP address terminate on the local router, regardless of whether that router is the primary VRRP router or a backup VRRP router. These packets include ping and Telnet traffic. Packets received on a Layer 2 (VLAN) interface destined for the VRRP virtual IP address terminate on the primary router.

VRRP Benefits

The benefits of VRRP are as follows:

- Redundancy—Enables you to configure multiple routers as the default gateway router, which reduces the possibility of a single point of failure in a network.

- Load sharing—Allows traffic to and from LAN clients to be shared by multiple routers. The traffic load is shared more equitably among available routers.
- Multiple VRRP groups—Supports multiple VRRP groups on a router physical interface if the platform supports multiple MAC addresses. Multiple VRRP groups enable you to implement redundancy and load sharing in your LAN topology.
- Multiple IP addresses—Allows you to manage multiple IP addresses, including secondary IP addresses. If you have multiple subnets that are configured on an Ethernet interface, you can configure VRRP on each subnet.
- Preemption—Enables you to preempt a backup router that has taken over for a failing primary with a higher priority backup router that has become available.
- Advertisement protocol—Uses a dedicated Internet Assigned Numbers Authority (IANA) standard multicast address (224.0.0.18) for VRRP advertisements. This addressing scheme minimizes the number of routers that must service the multicasts and allows test equipment to accurately identify VRRP packets on a segment. IANA has assigned the IP protocol number 112 to VRRP.
- VRRP tracking—Ensures that the best VRRP router is the primary for the group by altering VRRP priorities based on interface states.

Multiple VRRP Groups

You can configure multiple VRRP groups on a physical interface. For the number of supported VRRP groups, see the *Cisco Nexus® 3550-T Verified Scalability Guide*.

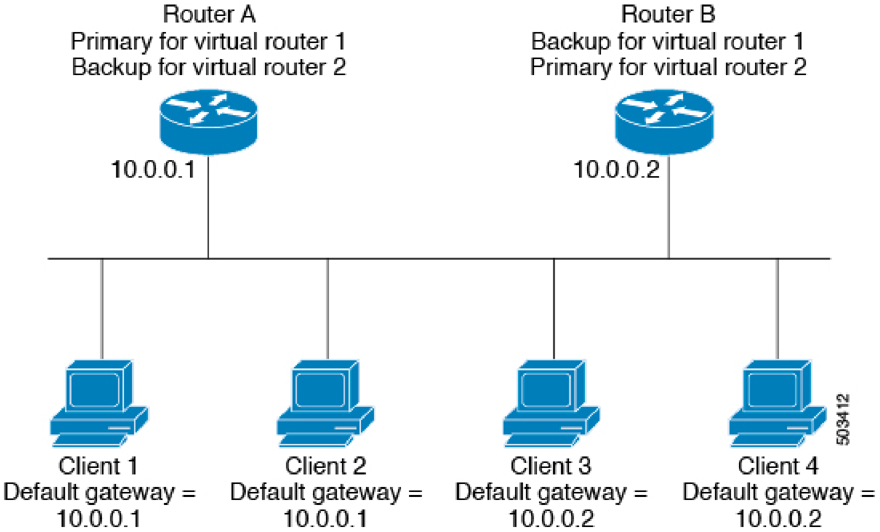
The number of VRRP groups that a router interface can support depends on the following factors:

- Router processing capability
- Router memory capability

In a topology where multiple VRRP groups are configured on a router interface, the interface can act as a primary for one VRRP group and as a backup for one or more other VRRP groups.

The following image shows a LAN topology in which VRRP is configured so that Routers A and B share the traffic to and from clients 1 through 4. Routers A and B act as backups to each other if either router fails.

Figure 21: Load Sharing and Redundancy VRRP Topology



This topology contains two virtual IP addresses for two VRRP groups that overlap. For VRRP group 1, Router A is the owner of IP address 10.0.0.1 and is the primary. Router B is the backup to Router A. Clients 1 and 2 are configured with the default gateway IP address of 10.0.0.1.

For VRRP group 2, Router B is the owner of IP address 10.0.0.2 and is the primary. Router A is the backup to router B. Clients 3 and 4 are configured with the default gateway IP address of 10.0.0.2.

VRRP Router Priority and Preemption

An important aspect of the VRRP redundancy scheme is the VRRP router priority because the priority determines the role that each VRRP router plays and what happens if the primary router fails.

If a VRRP router owns the virtual IP address and the IP address of the physical interface, this router functions as the primary. The priority of the primary is 255.

The priority also determines if a VRRP router functions as a backup router and the order of ascendancy to becoming a primary if the primary fails.

For example, if Router A, the primary in a LAN topology, fails, VRRP must determine if backups B or C should take over. If you configure Router B with priority 101 and Router C with the default priority of 100, VRRP selects Router B to become the primary because it has the higher priority. If you configure Routers B and C with the default priority of 100, VRRP selects the backup with the higher IP address to become the primary.

VRRP uses preemption to determine what happens after a VRRP backup router becomes the primary. With preemption enabled by default, VRRP switches to a backup if that backup comes online with a priority higher than the new primary. For example, if Router A is the primary and fails, VRRP selects Router B (next in order of priority). If Router C comes online with a higher priority than Router B, VRRP selects Router C as the new primary, even though Router B has not failed.

If you disable preemption, VRRP switches only if the original primary recovers or the new primary fails.

VRRP Advertisements

The VRRP primary sends VRRP advertisements to other VRRP routers in the same group. The advertisements communicate the priority and state of the primary. Cisco NX-OS encapsulates the VRRP advertisements in IP packets and sends them to the IP multicast address assigned to the VRRP group. Cisco NX-OS sends the advertisements once every second by default, but you can configure a different advertisement interval.

VRRP Authentication

VRRP supports the following authentication functions:

- No authentication
- Plain text authentication

VRRP rejects packets in any of the following cases:

- The authentication schemes differ on the router and in the incoming packet.
- Text authentication strings differ on the router and in the incoming packet.

VRRP Tracking

VRRP supports the following options for tracking:

- Native interface tracking—Tracks the state of an interface and uses that state to determine the priority of the VRRP router in a VRRP group. The tracked state is down if the interface is down or if the interface does not have a primary IP address.
- Object tracking—Tracks the state of a configured object and uses that state to determine the priority of the VRRP router in a VRRP group. See the *Configuring Object Tracking* section, for more information on object tracking.

If the tracked state (interface or object) goes down, VRRP updates the priority based on what you configure the new priority to be for the tracked state. When the tracked state comes up, VRRP restores the original priority for the virtual router group.

For example, you might want to lower the priority of a VRRP group member if its uplink to the network goes down so another group member can take over as primary for the VRRP group. See the Configuring VRRP Interface State Tracking, on page 371 section for more information.

Note VRRP does not support Layer 2 interface tracking.

High Availability

VRRP supports high availability through stateful restarts and stateful switchovers. A stateful restart occurs when the VRRP process fails and is restarted. A stateful switchover occurs when the active supervisor switches to the standby supervisor. Cisco NX-OS applies the run-time configuration after the switchover.

Guidelines and Limitations for VRRP

VRRP has the following configuration guidelines and limitations:

- You cannot configure VRRP on the management interface.
- When VRRP is enabled, you should replicate the VRRP configuration across devices in your network.
- We recommend that you do not configure more than one first-hop redundancy protocol on the same interface.
- You must configure an IP address for the interface on which you configure VRRP and enable that interface before VRRP becomes active.
- Cisco NX-OS removes all Layer 3 configurations on an interface when you change the interface VRF membership or the port channel membership or when you change the port mode to Layer 2.
- When you configure VRRP to track a Layer 2 interface, you must shut down the Layer 2 interface and reenable the interface to update the VRRP priority to reflect the state of the Layer 2 interface.

Default Settings for VRRP Parameters

The following table lists the default settings for VRRP parameters.

Table 21: Default VRRP Parameters

Parameters	Default
VRRP	Disabled
Advertisement interval	1 second
Authentication	No authentication
Preemption	Enabled
Priority	100

Configuring VRRP

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Enabling VRRP

You must globally enable VRRP before you configure and enable any VRRP groups.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **feature vrrp** **Example:** `switch(config)# feature vrrp`	Enables VRRP. Use the **no** form of this command to disable VRRP.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring VRRP Groups

You can create a VRRP group, assign the virtual IP address, and enable the group.

You can configure one virtual IPv4 address for a VRRP group. By default, the primary VRRP router drops the packets addressed directly to the virtual IP address because the VRRP primary is intended only as a next-hop router to forward packets. Some applications require that Cisco NX-OS accept packets that are addressed to the virtual router IP address. Use the secondary option to the virtual IP address to accept these packets when the local router is the VRRP primary.

Once you have configured the VRRP group, you must explicitly enable the group before it becomes active.

Before you begin

Ensure that you have configured an IP address on the interface. See Configuring IPv4 Addressing, on page 216.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *interface-type slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters interface configuration mode.

	Command or Action	Purpose

Step 3	**vrrp** *number*
	Example:
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		Creates a virtual router group. The range is 1–255.

Step 4	**address** *ip-address* [**secondary**] **Example:** `switch(config-if-vrrp)# address 192.0.2.8`	Configures the virtual IPv4 address for the specified VRRP group. This address should be in the same subnet as the IPv4 address of the interface. Use the **secondary** option only if applications require that VRRP routers accept the packets sent to the virtual router's IP address and deliver to applications.
Step 5	**no shutdown** **Example:** `switch(config-if-vrrp)# no shutdown`	Enables the VRRP group, which is disabled by default.
Step 6	(Optional) **show vrrp** **Example:** `switch(config-if-vrrp)# show vrrp`	Displays a summary of VRRP information.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-if-vrrp)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring VRRP Priority

The valid priority range for a virtual router is from 1 to 254 (1 is the lowest priority and 254 is the highest). The default priority value for backups is 100. For devices whose interface IP address is the same as the primary virtual IP address (the primary), the default value is 255.

Before you begin

Ensure that you have configured an IP address on the interface. See Configuring IPv4 Addressing, on page 216.

Ensure that you have enabled VRRP. (see the Configuring VRRP, on page 364 section).

Procedure

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

	Command or Action	Purpose

	`switch# configure terminal`
	`switch(config)#`	

Step 2	**interface** *interface-type slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**vrrp** *number*
	Example:
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		Creates a virtual router group.

Step 4	**shutdown** **Example:** `switch(config-if-vrrp)# shutdown`	Disables the VRRP group.
Step 5	**priority** *level* [**forwarding-threshold lower** *lower-value* **upper** *upper-value*] **Example:** `switch(config-if-vrrp)# priority 60 forwarding-threshold lower 40 upper 50`	Sets the priority level used to select the active router in a VRRP group. The *level* range is 1–254. The default is 100 for backups and 255 for a primary that has an interface IP address equal to the virtual IP address.
Step 6	**no shutdown** **Example:** `switch(config-if-vrrp)# no shutdown`	Enables the VRRP group.
Step 7	(Optional) **show vrrp** **Example:** `switch(config-if-vrrp)# show vrrp`	Displays a summary of VRRP information.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-if-vrrp)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring VRRP Authentication

You can configure simple text authentication for a VRRP group.

Before you begin

Ensure that you have configured an IP address on the interface (see Configuring IPv4 Addressing, on page 216).

Ensure that you have enabled VRRP (see the Configuring VRRP, on page 364 section).

Ensure that the authentication configuration is identical for all VRRP devices in the network.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *interface-type slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**vrrp** *number*
	Example:
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		Creates a virtual router group.

Step 4	**shutdown** **Example:** `switch(config-if-vrrp)# shutdown`	Disables the VRRP group.
Step 5	**authentication text** *password* **Example:** `switch(config-if-vrrp)# authentication text aPassword`	Assigns the simple text authentication option and specifies the keyname password. The keyname range is from 1 to 255 characters. We recommend that you use at least 16 characters. The text password is up to eight alphanumeric characters.
Step 6	**no shutdown** **Example:** `switch(config-if-vrrp)# no shutdown`	Enables the VRRP group, which is disabled by default.
Step 7	(Optional) **show vrrp** **Example:** `switch(config-if-vrrp)# show vrrp`	Displays a summary of VRRP information.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-if-vrrp)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Time Intervals for Advertisement Packets

You can configure the time intervals for advertisement packets.

Before you begin

Ensure that you have configured an IP address on the interface (see Configuring IPv4 Addressing, on page 216).

Ensure that you have enabled VRRP (see the Configuring VRRP, on page 364 section).

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *interface-type slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**vrrp** *number*
	Example:
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		Creates a virtual router group.

Step 4	**shutdown** **Example:** `switch(config-if-vrrp)# shutdown`	Disables the VRRP group.
Step 5	**advertisement interval** *seconds* **Example:** `switch(config-if-vrrp)# advertisement-interval 15`	Sets the interval time in seconds between sending advertisement frames. The range is from 1 to 255. The default is 1 second.
Step 6	**no shutdown** **Example:** `switch(config-if-vrrp)# no shutdown`	Enables the VRRP group.
Step 7	(Optional) **show vrrp** **Example:** `switch(config-if-vrrp)# show vrrp`	Displays a summary of VRRP information.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-if-vrrp)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Disabling Preemption

You can disable preemption for a VRRP group member. If you disable preemption, a higher-priority backup router does not take over for a lower-priority primary router. Preemption is enabled by default.

Before you begin

Ensure that you have configured an IP address on the interface. See Configuring IPv4 Addressing, on page 216.

Ensure that you have enabled VRRP. See the Configuring VRRP, on page 364 section.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *interface-type slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`    switch(config-if)#`
		Enters interface configuration mode.

Step 3	**vrrp** *number*
	Example:
	`switch(config-if)# vrrp 250`
	`    switch(config-if-vrrp)#`
		Creates a virtual router group.

Step 4	**shutdown** **Example:** `switch(config-if-vrrp)# shutdown`	Disables the VRRP group.
Step 5	**no preempt** **Example:** `switch(config-if-vrrp)# no preempt`	Disables the preempt option and allows the primary to remain when a higher-priority backup appears.
Step 6	**no shutdown** **Example:** `switch(config-if-vrrp)# no shutdown`	Enables the VRRP group.
Step 7	(Optional) **show vrrp** **Example:** `switch(config-if-vrrp)# show vrrp`	Displays a summary of VRRP information.

	Command or Action	Purpose
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-if-vrrp)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring VRRP Interface State Tracking

Interface state tracking changes the priority of the virtual router based on the state of another interface in the device. When the tracked interface goes down or the IP address is removed, Cisco NX-OS assigns the tracking priority value to the virtual router. When the tracked interface comes up and an IP address is configured on this interface, Cisco NX-OS restores the configured priority to the virtual router (see the Configuring VRRP Priority, on page 366 section).

Note VRRP does not support Layer 2 interface tracking.

Before you begin

Ensure that you have configured an IP address on the interface (see Configuring IPv4 Addressing, on page 216).

Ensure that you have enabled VRRP (see the Configuring VRRP, on page 364 section).

Ensure that you have enabled the virtual router (see the Configuring VRRP Groups, on page 365 section).

Ensure that you have enabled preemption on the interface.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *interface-type slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**vrrp** *number*
	Example:
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		Creates a virtual router group.

	Command or Action	Purpose
Step 4	**shutdown** **Example:** `switch(config-if-vrrp)# shutdown`	Disables the VRRP group.
Step 5	**track interface** *type slot/port* **priority** *value* **Example:** `switch(config-if-vrrp)# track interface ethernet 1/10 priority 254`	Enables interface priority tracking for a VRRP group. The priority range is from 1 to 254.
Step 6	**no shutdown** **Example:** `switch(config-if-vrrp)# no shutdown`	Enables the VRRP group.
Step 7	(Optional) **show vrrp** **Example:** `switch(config-if-vrrp)# show vrrp`	Displays a summary of VRRP information.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-if-vrrp)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring VRRP Object Tracking

You can track an IPv4 object using VRRP.

Before you begin

Make sure that VRRP is enabled.

Configure object tracking using the commands in the *Configuring Object Tracking* section.

Procedure

	Command or Action	Purpose

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

Step 2	**interface type number**
	Example:
	`switch(config)#`
	`switch(config-if)# interface ethernet 1/1`
	`switch(config-if)#`
		Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose

Step 3	**vrrp** *number* **address-family ipv4**
	Example:
	`switch(config-if)# vrrp 5 address-family ipv4`
	`switch(config-if-vrrp-group)#`
		Creates a VRRP group for IPv4 and enters VRRP vrrp number address-family ipv4 group configuration mode. The range is from 1 to 255.

Step 4	**track** *object-number* **decrement** *number* **Example:** `switch(config-if-vrrp-group)# track 1 decrement 2`	Creates a virtual router group. The range is from 1 to 255.
Step 5	(Optional) **show running-config vrrp** **Example:** `switch(config-if-vrrp-group)# show running-config vrrp`	Displays the running configuration for VRRP.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if-vrrp-group)# copy running-config startup-config`	Saves this configuration change.

Verifying the VRRP Configuration

To display VRRP configuration information, perform one of the following tasks:

Command	Purpose
show interface *interface-type*	Displays the virtual router configuration for an interface.
show fhrp *interface-type interface-number*	Displays First Hop Redundancy Protocol (FHRP) information.
show vrrp [*group-number*]	Displays the VRRP status for all groups or for a specific VRRP group.

Monitoring and Clearing VRRP Statistics

To display VRRP statistics, use the following commands:

Command	Purpose
show vrrp statistics	Displays the VRRP statistics.

Use the **clear vrrp statistics** command to clear the VRRP statistics for all interfaces on the device.

Configuration Examples for VRRP

In this example, Router A and Router B each belong to three VRRP groups. In the configuration, each group has the following properties:

- Group 1:
  - Virtual IP address is 10.1.0.10.
  - Router A becomes the primary for this group with priority 120.
  - Advertising interval is 3 seconds.
  - Preemption is enabled.
- Group 5:
  - Router B becomes the primary for this group with priority 200.
  - Advertising interval is 30 seconds.
  - Preemption is enabled.
- Group 100:
  - Router A becomes the primary for this group first because it has a higher IP address (10.1.0.2).
  - Advertising interval is the default of 1 second.
  - Preemption is disabled.

Router A

```
switch (config)# interface ethernet 1/0
switch (config-if)# ip address 10.1.0.2/16
switch (config-if)# no shutdown
switch (config-if)# vrrp 1
switch (config-if-vrrp)# priority 120
switch (config-if-vrrp)# authentication text cisco
switch (config-if-vrrp)# advertisement-interval 3
switch (config-if-vrrp)# address 10.1.0.10
switch (config-if-vrrp)# no shutdown
switch (config-if-vrrp)# exit
switch (config-if)# vrrp 5
switch (config-if-vrrp)# priority 100
switch (config-if-vrrp)# advertisement-interval 30
switch (config-if-vrrp)# address 10.1.0.50
switch (config-if-vrrp)# no shutdown
switch (config-if-vrrp)# exit
switch (config-if)# vrrp 100
switch (config-if-vrrp)# no preempt
switch (config-if-vrrp)# address 10.1.0.100
switch (config-if-vrrp)# no shutdown
```

Router B

```
switch (config)# interface ethernet 1/0
switch (config-if)# ip address 10.2.0.1/2
switch (config-if)# no shutdown
switch (config-if)# vrrp 1
switch (config-if-vrrp)# priority 100
switch (config-if-vrrp)# authentication text cisco
switch (config-if-vrrp)# advertisement-interval 3
switch (config-if-vrrp)# address 10.2.0.10
switch (config-if-vrrp)# no shutdown
switch (config-if-vrrp)# exit
switch (config-if)# vrrp 5
switch (config-if-vrrp)# priority 200
switch (config-if-vrrp)# advertisement-interval 30
switch (config-if-vrrp)# address 10.2.0.50
switch (config-if-vrrp)# no shutdown
switch (config-if-vrrp)# exit
switch (config-if)# vrrp 100
switch (config-if-vrrp)# no preempt
switch (config-if-vrrp)# address 10.2.0.100
switch (config-if-vrrp)# no shutdown
```

Additional References

Related Documents for VRRP

Related Topic	Document Title
Configuring high availability	*Cisco Nexus® High Availability and Redundancy Guide*

PART VI

Cisco Nexus 3550-T Layer 2 Switching Configuration Guide

CHAPTER 24

Layer 2 Switching Configuration Guide

This preface includes the following sections:

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Layer 2 Ethernet Switching Overview

Information About Layer 2 Switching

Note See the *Cisco Nexus® 3550-T Interfaces Configuration* section, for information on creating interfaces.

You can configure Layer 2 switching ports as access or trunk ports. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network. All Layer 2 switching ports maintain MAC address tables.

Layer 2 Ethernet Switching Overview

The device supports simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

The device solves congestion problems caused by high-bandwidth devices and a large number of users by assigning each device (for example, a server) to its own collision domain. Because each LAN port connects to a separate Ethernet collision domain, servers in a switched environment achieve full access to the bandwidth.

Because collisions cause significant congestion in Ethernet networks, an effective solution is full-duplex communication. In full-duplex mode, which is configurable on these interfaces, two stations can transmit and receive at the same time. When packets can flow in both directions simultaneously, the effective Ethernet bandwidth doubles.

Each LAN port on a device can connect to a single workstation, server, or to another device through which workstations or servers connect to the network.

To reduce signal degradation, the device considers each LAN port to be an individual segment. When stations connected to different LAN ports need to communicate, the device forwards frames from one LAN port to the other at wire speed to ensure that each session receives full bandwidth.

To switch frames between LAN ports efficiently, the device maintains an address table. When a frame enters the device, it associates the media access control (MAC) address of the sending network device with the LAN port on which it was received.

The device dynamically builds the address table by using the MAC source address of the frames received. When the device receives a frame for a MAC destination address not listed in its address table, it floods the frame to all LAN ports of the same VLAN except the port that received the frame. When the destination station replies, the device adds its relevant MAC source address and port ID to the address table. The device then forwards subsequent frames to a single LAN port without flooding all LAN ports.

You can configure MAC addresses, which are called static MAC addresses, to statically point to specified interfaces on the device. These static MAC addresses override any dynamically learned MAC addresses on those interfaces. You cannot configure broadcast addresses as static MAC addresses. The static MAC entries are retained across a reboot of the device.

The address table can store a number of MAC address entries depending on the hardware I/O module. The device uses an aging mechanism, defined by a configurable aging timer, so if an address remains inactive for a specified number of seconds, it is removed from the address table.

Layer 3 Static MAC Addresses

You can configure a static MAC address for the following Layer 3 interfaces:

- Layer 3 interfaces
- Layer 3 port channels
- VLAN network interface

See the *Cisco Nexus® 3550-T Interfaces Configuration* section, for information on configuring Layer 3 interfaces.

VLANs

A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment.

Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered as a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a bridge or a router.

All ports are assigned to the default VLAN (VLAN1) when the device first comes up. A VLAN interface, or switched virtual interface (SVI), is a Layer 3 interface that is created to provide communication between VLANs.

The devices support 4095 VLANs range (255 maximum VLANs supported) in accordance with the IEEE 802.1Q standard. These VLANs are organized into several ranges, and you use each range slightly differently. Some of these VLANs are reserved for internal use by the device and are not available for configuration.

Note Inter-Switch Link (ISL) trunking is not supported on the Cisco NX-OS.

Spanning Tree

This section discusses the implementation of the Spanning Tree Protocol (STP) on the software. Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. When the IEEE 802.1D Spanning Tree Protocol is referred to in the publication, 802.1D is stated specifically.

STP Overview

STP provides a loop-free network at the Layer 2 level. Layer 2 LAN ports send and receive STP frames, which are called Bridge Protocol Data Units (BPDUs), at regular intervals. Network devices do not forward these frames but use the frames to construct a loop-free path.

802.1D is the original standard for STP, and many improvements have enhanced the basic loop-free STP. Additionally, the entire standard was reworked to make the loop-free convergence process faster to keep up with the faster equipment.

Finally, the 802.1s standard, Multiple Spanning Trees (MST), allows you to map multiple VLANs into a single spanning tree instance. Each instance runs an independent spanning tree topology.

Although the software can interoperate with legacy 802.1D systems, the system runs MST. MST is the default STP protocol for Cisco Nexus devices.

Note Cisco NX-OS uses the extended system ID and MAC address reduction; you cannot disable these features.

In addition, Cisco has created some proprietary features to enhance the spanning tree activities.

MST

MST is the default spanning tree mode for the software and is enabled by default on the default VLAN and all newly created VLANs.

The multiple independent spanning tree topologies enabled by MST provide multiple forwarding paths for data traffic, enable load balancing, and reduce the number of STP instances required to support a large number of VLANs.

MST incorporates RSTP, so it also allows rapid convergence. MST improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

You can force specified interfaces to send prestandard, rather than standard, MST messages using the command-line interface.

STP Extensions

The software supports the following Cisco proprietary features:

- Spanning tree port types—The default spanning tree port type is normal. You can configure interfaces connected to Layer 2 hosts as edge ports and interfaces connected to Layer 2 switches or bridges as network ports.
- BPDU Guard—BPDU Guard shuts down the port if that port receives a BPDU.
- BPDU Filter—BPDU Filter suppresses sending and receiving BPDUs on the port.
- Loop Guard—Loop Guard helps prevent bridging loops that could occur because of a unidirectional link failure on a point-to-point link.
- Root Guard—STP root guard prevents a port from becoming root port or blocked port. If a port configured for root guard receives a superior BPDU, the port immediately goes to the root-inconsistent (blocked) state.

About Traffic Storm Control

The traffic storm control threshold numbers and the time interval allow the traffic storm control algorithm to work with different levels of granularity. A higher threshold allows more packets to pass through.

Traffic storm control on the Cisco Nexus 3550-T device is implemented in the hardware. The traffic storm control circuitry monitors packets that pass from a Layer 2 interface to the switching bus.

Related Topics

The following documents are related to the Layer 2 switching features:

- *Cisco Nexus® 3550-T Interfaces Configuration section*
- *Cisco Nexus® 3550-T Security Configuration section*
- *Cisco Nexus® 3550-T System Management Configuration section*

CHAPTER 25

Configuring Layer 2 Switching

Information About Layer 2 Switching

Note See the , for information on creating interfaces.

You can configure Layer 2 switching ports as access or trunk ports. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network. All Layer 2 switching ports maintain MAC address tables.

Layer 2 Ethernet Switching Overview

The device supports simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

The device solves congestion problems caused by high-bandwidth devices and a large number of users by assigning each device (for example, a server) to its own collision domain. Because each LAN port connects to a separate Ethernet collision domain, servers in a switched environment achieve full access to the bandwidth.

Because collisions cause significant congestion in Ethernet networks, an effective solution is full-duplex communication. In full-duplex mode, which is configurable on these interfaces, two stations can transmit and receive at the same time. When packets can flow in both directions simultaneously, the effective Ethernet bandwidth doubles.

Switching Frames Between Segments

Each LAN port on a device can connect to a single workstation, server, or to another device through which workstations or servers connect to the network.

To reduce signal degradation, the device considers each LAN port to be an individual segment. When stations connected to different LAN ports need to communicate, the device forwards frames from one LAN port to the other at wire speed to ensure that each session receives full bandwidth.

To switch frames between LAN ports efficiently, the device maintains an address table. When a frame enters the device, it associates the media access control (MAC) address of the sending network device with the LAN port on which it was received.

Building the Address Table and Address Table Changes

The device dynamically builds the address table by using the MAC source address of the frames received. When the device receives a frame for a MAC destination address not listed in its address table, it floods the frame to all LAN ports of the same VLAN except the port that received the frame. When the destination station replies, the device adds its relevant MAC source address and port ID to the address table. The device then forwards subsequent frames to a single LAN port without flooding all LAN ports.

You can configure MAC addresses, which are called static MAC addresses, to statically point to specified interfaces on the device. These static MAC addresses override any dynamically learned MAC addresses on those interfaces. You cannot configure broadcast addresses as static MAC addresses. The static MAC entries are retained across a reboot of the device.

The address table can store a number of MAC address entries depending on the hardware I/O module. The device uses an aging mechanism, defined by a configurable aging timer, so if an address remains inactive for a specified number of seconds, it is removed from the address table.

Layer 3 Static MAC Addresses

You can configure a static MAC address for the following Layer 3 interfaces:

- Layer 3 interfaces
- Layer 3 port channels
- VLAN network interface

Note You cannot configure static MAC address on tunnel interfaces.

See the Cisco Nexus Series NX-OS Interfaces Configuration Guide, for information on configuring Layer 3 interfaces.

Prerequisites for Configuring MAC Addresses

MAC addresses have the following prerequisites:

- You must be logged onto the device.
- If necessary, install the Advanced Services license.

Default Settings for Layer 2 Switching

This table lists the default setting for Layer 2 switching parameters.

Table 22: Default Layer 2 Switching Parameters

Parameters	Default
Aging time	1800 seconds

Configuring Layer 2 Switching by Steps

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Configuring a Static MAC Address

You can configure MAC addresses, which are called static MAC addresses, to statically point to specified interfaces on the device. These static MAC addresses override any dynamically learned MAC addresses on those interfaces. You cannot configure broadcast or multicast addresses as static MAC addresses.

Procedure

	Command or Action	Purpose

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

Step 2	**mac address-table static** *mac-address* **vlan** *vlan-id* {[**drop** | **interface** {*type slot/port*} | **port-channel** *number*]}
	Example:
	`switch(config)# mac address-table static 1.1.1 vlan 2 interface ethernet 1/2`
		Specifies a static MAC address to add to the Layer 2 MAC address table.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits the configuration mode.

	Command or Action	Purpose
Step 4	(Optional) **show mac address-table static** **Example:** `switch# show mac address-table static`	Displays the static MAC addresses.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to put a static entry in the Layer 2 MAC address table:

```
switch# config t
switch(config)# mac address-table static 1.1.1 vlan 2 interface ethernet 1/2
switch(config)#
```

Configuring a Static MAC Address on a Layer 3 Interface

You can configure static MAC addresses on Layer 3 interfaces. You cannot configure broadcast or multicast addresses as static MAC addresses.

See the *Configuring Layer 3 Interfaces* section, for information on configuring Layer 3 interfaces.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** [**ethernet** *slot/port* | **ethernet** *slot/port.number* | **port-channel** *number* | **vlan** *vlan-id*] **Example:** `switch(config)# interface ethernet 1/3`	Specifies the Layer 3 interface and enters the interface configuration mode. **Note** You must create the Layer 3 interface before you can assign the static MAC address.

Step 3	**mac-address** *mac-address*
	Example:
	`switch(config-if)# mac-address 22ab.47dd.ff89`
	`switch(config-if)#`
		Specified a static MAC address to add to the Layer 3 interface.

	Command or Action	Purpose

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 5	(Optional) **show interface** [**ethernet** *slot/port* | **ethernet** *slot/port.number* | **port-channel** *number* | **vlan** *vlan-id*] **Example:** `switch# show interface ethernet 1/3`	Displays information about the Layer 3 interface.

Step 6	(Optional) **copy running-config startup-config**
	Example:
	`switch# copy running-config`
	`startup-config`
		Copies the running configuration to the startup configuration.

Example

This example shows how to configure the Layer 3 interface on slot 1, port 3 with a static MAC address:

```
switch# config t
switch(config)# interface ethernet 1/3
switch(config-if)# mac-address 22ab.47dd.ff89
switch(config-if)#
```

Configuring the Aging Time for the MAC Table

You can configure the amount of time that a MAC address entry (the packet source MAC address and port on which that packet was learned) remains in the MAC table, which contains the Layer 2 information.

Note MAC addresses are aged out up to two times the configured MAC address table aging timeout.

Note You can also configure the MAC aging time in interface configuration mode or VLAN configuration mode.

Procedure

	Command or Action	Purpose

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

Step 2	**mac address-table aging-time** *seconds* **Example:** `switch(config)# mac address-table aging-time 600`	Specifies the time before an entry ages out and is discarded from the Layer 2 MAC address table. The range is from 120 to 918000; the default is 1800 seconds. Entering the value 0 disables the MAC aging.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits the configuration mode.

Step 4	(Optional) **show mac address-table aging-time** **Example:** `switch# show mac address-table aging-time`	Displays the aging time configuration for MAC address retention.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to set the ageout time for entries in the Layer 2 MAC address table to 600 seconds (10 minutes):

```
switch# config t
switch(config)# mac address-table aging-time 600
switch(config)#
```

Clearing Dynamic Addresses from the MAC Table

You can clear all dynamic Layer 2 entries in the MAC address table. (You can also clear entries by designated interface or VLAN.)

Procedure

	Command or Action	Purpose

Step 1	**clear mac address-table dynamic** {**address** *mac_addr*} {**interface** [**ethernet** *slot/port* | **port-channel** *channel-number*]} {**vlan** *vlan_id*}
	Example:
	`switch# clear mac address-table dynamic`
		Clears the dynamic address entries from the MAC address table in Layer 2.

Step 2	(Optional) **show mac address-table** **Example:** `switch# show mac address-table`	Displays the MAC address table.

Example

This example shows how to clear the dynamic entries in the Layer 2 MAC address table:

```
switch# clear mac address-table dynamic
switch#
```

Verifying the Layer 2 Switching Configuration

To display Layer 2 switching configuration information, perform one of the following tasks:

Command	Purpose
show mac address-table	Displays information about the MAC address table.
show mac address-table aging-time	Displays information about the aging time set for the MAC address entries.
show mac address-table static	Displays information about the static entries on the MAC address table.
show interface [*interface*] **mac-address**	Displays the MAC addresses and the burn-in MAC address for the interfaces.

Configuration Example for Layer 2 Switching

The following example shows how to add a static MAC address and how to modify the default global aging time for MAC addresses:

```
switch# configure terminal
switch(config)# mac address-table static 0000.0000.1234 vlan 10 interface ethernet 1/15
switch(config)# mac address-table aging-time 120
```

Additional References for Layer 2 Switching -- CLI Version

Related Documents

Related Topic	Document Title
Static MAC addresses	*Cisco Nexus® 3550-T Security Configuration* section
Interfaces	*Cisco Nexus® 3550-T Interfaces Configuration* section
System management	*Cisco Cisco Nexus® 3550-T System Management Configuration* section

CHAPTER 26

Configuring MST Using Cisco NX-OS

Information About MST

Note See the *Cisco Nexus® 3550-T Interfaces Configuration* section, for information on creating Layer 2 interfaces.

MST, which is the IEEE 802.1s standard, allows you to assign two or more VLANs to a spanning tree instance. MST is not the default spanning tree mode; Rapid per VLAN Spanning Tree (Rapid PVST+) is the default mode. MST instances with the same name, revision number, and VLAN-to-instance mapping combine to form an MST region. The MST region appears as a single bridge to spanning tree configurations outside the region. MST forms a boundary to that interface when it receives an IEEE 802.1D Spanning Tree Protocol (STP) message from a neighboring device.

Note Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the IEEE 802.1D Spanning Tree Protocol is discussed in this publication, 802.1D is stated specifically.

MST Overview

Note MST is the default spanning tree mode.

MST provides rapid convergence through explicit handshaking because each MST instance uses the IEEE 802.1w standard, which eliminates the 802.1D forwarding delay and quickly transitions root bridge ports and designated ports to the forwarding state.

MAC address reduction is always enabled on the device. You cannot disable this feature.

MST improves spanning tree operation and maintains backward compatibility with original 802.1D spanning tree STP versions:

Note

- IEEE 802.1 was defined in the Rapid Spanning Tree Protocol (RSTP) and was incorporated into IEEE 802.1D.
- IEEE 802.1 was defined in MST and was incorporated into IEEE 802.1Q

  .

MST Regions

To allow devices to participate in MST instances, you must consistently configure the devices with the same MST configuration information.

A collection of interconnected devices that have the same MST configuration is an MST region. An MST region is a linked group of MST bridges with the same MST configuration.

The MST configuration controls the MST region to which each device belongs. The configuration includes the name of the region, the revision number, and the VLAN-to-MST instance assignment mapping.

A region can have one or multiple members with the same MST configuration. Each member must be capable of processing 802.1w bridge protocol data units (BPDUs). There is no limit to the number of MST regions in a network.

Each device can support only single MST instance (Instance 0), in a single MST region. You can assign a VLAN to only one MST instance at a time.

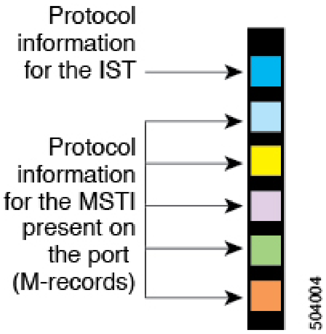
The MST region appears as a single bridge to adjacent MST regions and to other 802.1D spanning tree protocols.

Note We do not recommend that you partition the network into a large number of regions.

MST BPDUs

Each device has only one MST BPDU per interface, and that BPDU carries an M-record for each MSTI on the device. Only the IST sends BPDUs for the MST region; all M-records are encapsulated in that one BPDU that the IST sends. Because the MST BPDU carries information for all instances, the number of BPDUs that need to be processed to support MST is significantly reduced.

Figure 22: MST BPDU with M-Records for MSTIs


Protocol information for the IST
Protocol information for the MSTI present on the port (M-records)
504004

MST Configuration Information

The MST configuration that must be identical on all devices within a single MST region is configured by the user.

You can configure the three parameters of the MST configuration as follows:

- Name—32-character string, null padded and null terminated, identifying the MST region
- Revision number—Unsigned 16-bit number that identifies the revision of the current MST configuration

Note You must set the revision number when required as part of the MST configuration. The revision number is not incremented automatically each time that the MST configuration is committed.

- VLAN-to-MST instance mapping—4096-element table that associates each of the potential VLANs supported to a given instance with the first (0) and last element (4095) set to 0. The value of element number X represents the instance to which VLAN X is mapped.

Note When you change the VLAN-to-MSTI mapping, the system reconverges MST.

MST BPDUs contain these three configuration parameters. An MST bridge accepts an MST BPDU into its own region only if these three configuration parameters match exactly. If one configuration attribute differs, the MST bridge considers the BPDU to be from another MST region.

IST, CIST, and CST

IST, CIST, and CST Overview

MST establishes and maintains IST, CIST, and CST spanning trees, as follows:

- An IST is the spanning tree that runs in an MST region.

MST establishes and maintains additional spanning trees within each MST region; these spanning trees are called multiple spanning tree instances (MSTIs).

Instance 0 is a special instance for a region, known as the IST. The IST always exists on all ports; you cannot delete the IST, or Instance 0. By default, all VLANs are assigned to the IST. All other MST instances are numbered from 1 to 4094.

The IST is the only STP instance that sends and receives BPDUs. All of the other MSTI information is contained in MST records (M-records), which are encapsulated within MST BPDUs.

All MSTIs within the same region share the same protocol timers, but each MSTI has its own topology parameters, such as the root bridge ID, the root path cost, and so forth.

An MSTI is local to the region; for example, MSTI 9 in region A is independent of MSTI 9 in region B, even if regions A and B are interconnected. Only CST information crosses region boundaries.

- The CST interconnects the MST regions and any instance of 802.1D and 802.1w STP that may be running on the network. The CST is the one STP instance for the entire bridged network and encompasses all MST regions and 802.1w and 802.1D instances.
- A CIST is a collection of the ISTs in each MST region. The CIST is the same as an IST inside an MST region, and the same as a CST outside an MST region.

The spanning tree computed in an MST region appears as a subtree in the CST that encompasses the entire switched domain. The CIST is formed by the spanning tree algorithm running among devices that support the 802.1w, 802.1s, and 802.1D standards. The CIST inside an MST region is the same as the CST outside a region.

Spanning Tree Operation Within an MST Region

The IST connects all the MSTdevices in a region. When the IST converges, the root of the IST becomes the CIST regional root. The CIST regional root is also the CIST root if there is only one region in the network. If the CIST root is outside the region, the protocol selects one of the MST devices at the boundary of the region as the CIST regional root.

When an MST device initializes, it sends BPDUs that identify itself as the root of the CIST and the CIST regional root, with both the path costs to the CIST root and to the CIST regional root set to zero. The device also initializes all of its MSTIs and claims to be the root for all of them. If the device receives superior MSTI root information (lower switch ID, lower path cost, and so forth) than the information that is currently stored for the port, it relinquishes its claim as the CIST regional root.

During initialization, an MST region might have many subregions, each with its own CIST regional root. As devices receive superior IST information from a neighbor in the same region, they leave their old subregions and join the new subregion that contains the true CIST regional root. This action causes all subregions to shrink except for the subregion that contains the true CIST regional root.

All devices in the MST region must agree on the same CIST regional root. Any two devices in the region will only synchronize their port roles for an MSTI if they converge to a common CIST regional root.

Spanning Tree Operations Between MST Regions

If you have multiple regions or 802.1 w or 802.1D STP instances within a network, MST establishes and maintains the CST, which includes all MST regions and all 802.1w and 802.1D STP devices in the network. The MSTIs combine with the IST at the boundary of the region to become the CST.

The IST connects all the MST devices in the region and appears as a subtree in the CIST that encompasses the entire switched domain. The root of the subtree is the CIST regional root. The MST region appears as a virtual device to adjacent STP devices and MST regions.

MST Terminology

MST naming conventions include identification of some internal or regional parameters. These parameters are used only within an MST region, compared to external parameters that are used throughout the whole network. Because the CIST is the only spanning tree instance that spans the whole network, only the CIST parameters require the external qualifiers and not the internal or regional qualifiers. The MST terminology is as follows:

- The CIST root is the root bridge for the CIST, which is the unique instance that spans the whole network.
- The CIST external root path cost is the cost to the CIST root. This cost is left unchanged within an MST region. An MST region looks like a single device to the CIST. The CIST external root path cost is the root path cost calculated between these virtual devices and devices that do not belong to any region.
- If the CIST root is in the region, the CIST regional root is the CIST root. Otherwise, the CIST regional root is the closest device to the CIST root in the region. The CIST regional root acts as a root bridge for the IST.
- The CIST internal root path cost is the cost to the CIST regional root in a region. This cost is only relevant to the IST, instance 0.

Hop Count

MST does not use the message-age and maximum-age information in the configuration BPDU to compute the STP topology inside the MST region. Instead, the protocol uses the path cost to the root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism.

By using the **spanning-tree mst max-hops** global configuration command, you can configure the maximum hops inside the region and apply it to the IST and all MST instances in that region.

The hop count achieves the same result as the message-age information (triggers a reconfiguration). The root bridge of the instance always sends a BPDU (or M-record) with a cost of 0 and the hop count set to the maximum value. When a device receives this BPDU, it decrements the received remaining hop count by one and propagates this value as the remaining hop count in the BPDUs that it generates. When the count reaches zero, the device discards the BPDU and ages the information held for the port.

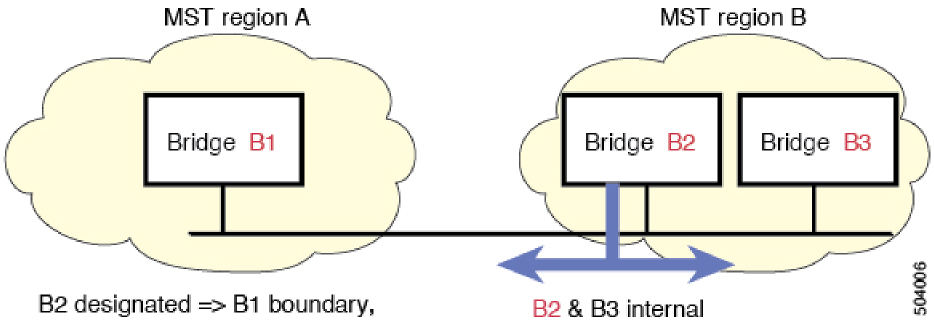
The message-age and maximum-age information in the 802.1w portion of the BPDU remain the same throughout the region (only on the IST), and the same values are propagated by the region-designated ports at the boundary.

You configure a maximum aging time as the number of seconds that a device waits without receiving spanning tree configuration messages before attempting a reconfiguration.

Boundary Ports

A boundary port is a port that connects to a LAN, the designated bridge of a bridge with a different MST configuration (and so, a separate MST region) 802.1D STP bridge. A designated port knows that it is on the boundary if it detects an STP bridge or receives an agreement proposal from an MST bridge with a different configuration. This definition allows two ports that are internal to a region to share a segment with a port that belongs to a different region, creating the possibility of receiving both internal and external messages on a port.

Figure 23: MST Boundary Ports



At the boundary, the roles of MST ports do not matter; the system forces their state to be the same as the IST port state. If the boundary flag is set for the port, the MST port-role selection process assigns a port role to the boundary and assigns the same state as the state of the IST port. The IST port at the boundary can take up any port role except a backup port role.

Port Cost and Port Priority

Spanning tree uses port costs to break a tie for the designated port. Lower values indicate lower port costs, and spanning tree chooses the least costly path. Default port costs are taken from the bandwidth of the interface, as follows:

- 1 Gigabit Ethernet—20,000
- 10 Gigabit Ethernet—2,000
- 40 Gigabit Ethernet—500

You can configure the port costs in order to influence which port is chosen.

Note MST always uses the long path-cost calculation method, so the range of valid values is between 1 and 200,000,000.

The system uses port priorities to break ties among ports with the same cost. A lower number indicates a higher priority. The default port priority is 128. You can configure the priority to values between 0 and 224, in increments of 32.

Interoperability with IEEE 802.1D

A device that runs MST supports a built-in protocol migration feature that enables it to interoperate with 802.1D STP devices. If this device receives an 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. In addition, an MST device can detect that a port is at the boundary of a region when it receives an 802.1D BPDU, an MST BPDU (Version 3) associated with a different region, or an 802.1w BPDU (Version 2).

However, the device does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D device has been removed from the link unless the 802.1D device is the designated device. A device might also continue to assign a boundary role to a port when the device to which this device is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring devices), enter the **clear spanning-tree detected-protocols** command.

All 8021.D STP switches on the link can process MST BPDUs as if they are 802.1w BPDUs. MST devices can send either Version 0 configuration and topology change notification (TCN) BPDUs or Version 3 MST BPDUs on a boundary port. A boundary port connects to a LAN, the designated device of which is either a single spanning tree device or a device with a different MST configuration.

MST interoperates with the Cisco prestandard MSTP whenever it receives prestandard MSTP on an MST port; no explicit configuration is necessary.

You can also configure the interface to proactively send prestandard MSTP messages.

High Availability for MST

The software supports high availability for MST. However, the statistics and timers are not restored when MST restarts. The timers start again and the statistics begin from 0.

Prerequisites for MST

MST has the following prerequisites:

- You must be logged onto the device.

Guidelines and Limitations for Configuring MST

Note When you change the VLAN-to-MSTI mapping, the system reconverges MST.

MST has the following configuration guidelines and limitations:

- For MST configuration limits, see the *Cisco Nexus® 3550-T Verified Scalability Guide*.
- **show** commands with the **internal** keyword are not supported.
- MST is the default spanning tree mode.
- You can assign a VLAN to only one MST instance in Cisco Nexus® 3550-T switches.
- By default, all VLANs are mapped to MSTI 0 or the IST.
- You can load balance only within the MST region.
- Ensure that trunks within an MST region carry all of the VLANs that are mapped to an MSTI or exclude all those VLANs that are mapped to an MSTI.
- Always leave STP enabled.

- Do not change timers because you can adversely affect your network stability.
- Keep user traffic off the management VLAN; keep the management VLAN separate from user data.
- Choose the distribution and core layers as the location of the primary and secondary root switches.
- Port channeling—The port channel bundle is considered as a single port. The port cost is the aggregation of all the configured port costs assigned to that channel.
- When you map a VLAN to an MSTI, the system automatically removes that VLAN from its previous MSTI.
- You can map any number of VLANs to an MSTI.
- Do not partition the network into a large number of regions. However, if this situation is unavoidable, we recommend that you partition the switched LAN into smaller LANs interconnected by non-Layer 2 devices.
- When you are in the MST configuration submode, the following guidelines apply:
    - Each command reference line creates its pending regional configuration.
    - The pending region configuration starts with the current region configuration.
    - To leave the MST configuration submode without committing any changes, enter the **abort** command.
    - To leave the MST configuration submode and commit all the changes that you made before you left the submode, enter the **exit** or **end** commands, or press **Ctrl + Z**.

Default Settings for MST

This table lists the default settings for MST parameters.

Table 23: Default MST Parameters

Parameters	Default
Spanning tree	Enabled
Name	Empty string
VLAN mapping	All VLANs mapped to a CIST instance
Revision	0
Instance ID	Instance 0; VLANs 1 to 3967 are mapped to Instance 0 by default
MSTI per MST region	Only single instance of MST is permitted in the Cisco Nexus® 3550-T switches
Bridge priority (configurable per CIST port)	32768
Spanning tree port priority (configurable per CIST port)	128

Parameters	Default
Spanning tree port cost (configurable per CIST port)	Auto The default port cost is determined by the port speed as follows: • 1 Gigabit Ethernet: 20,000 • 10 Gigabit Ethernet: 2,000 • 40 Gigabit Ethernet: 500
Hello time	2 seconds
Forward-delay time	15 seconds
Maximum-aging time	20 seconds
Maximum hop count	20 hops
Link type	Auto The default link type is determined by the duplex, as follows: • Full duplex: point-to-point link • Half duplex: shared link

Configuring MST

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco software commands for this feature might differ from the Cisco IOS commands that you would use.

Enabling MST - CLI Version

MST is the default spanning tree mode.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree mode mst**. **Example:**	• **spanning-tree mode mst** Enables MST on the device.

	Command or Action	Purpose
	`switch(config)# spanning-tree mode mst`	

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show running-config spanning-tree all** **Example:** `switch# show running-config spanning-tree all`	Displays the currently running STP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enable MST on the device:

```
switch# config t
switch(config)# spanning-tree mode mst
switch(config)# exit
switch#
```

Entering MST Configuration Mode

You enter MST configuration mode to configure the MST name, VLAN-to-instance mapping, and MST revision number on the device.

If two or more devices are in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

Note Each command reference line creates its pending regional configuration in MST configuration mode. In addition, the pending region configuration starts with the current region configuration.

Procedure

	Command or Action	Purpose
Step 1	**config t** **Example:**	Enters configuration mode.

	Command or Action	Purpose

	`switch# config t`
	`switch(config)#`	

Step 2	**spanning-tree mst configuration** or **no spanning-tree mst configuration**
	Example:
	`switch(config)# spanning-tree mst configuration`
	`switch(config-mst)#`
		• **spanning-tree mst configuration**
		Enters MST configuration submode on the system. You must be in the MST configuration submode to assign the MST configuration parameters, as follows:
		• MST name
		• VLAN-to-MST instance mapping
		• MST revision number
		• **no spanning-tree mst configuration**
		Returns the MST region configuration to the following default values:
		• The region name is an empty string.
		• No VLANs are mapped to any MST instance (all VLANs are mapped to the CIST instance).
		• The revision number is 0.

Step 3	**exit** or **abort**
	Example:
	`switch(config-mst)# exit`
	`switch(config)#`
		• **exit**
		Commits all the changes and exits MST configuration submode.
		• **abort**
		Exits the MST configuration submode without committing any of the changes.

Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enter the MST configuration submode on the device:

```
switch# config t
switch(config)# spanning-tree mst configuration
switch(config-mst)# exit
switch(config)#
```

Specifying the MST Name

You can configure a region name on the bridge. If two or more bridges are in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree mst configuration**
	Example:
	`switch(config)# spanning-tree mst configuration`
	`switch(config-mst)#`
		Enters MST configuration submode.

Step 3	**name** *name* **Example:** `switch(config-mst)# name accounting`	Specifies the name for the MST region. The *name* string has a maximum length of 32 characters and is case sensitive. The default is an empty string.

Step 4	**exit** or **abort**
	Example:
	`switch(config-mst)# exit`
	`switch(config)#`
		• **exit**
		Commits all the changes and exits MST configuration submode.
		• **abort**
		Exits the MST configuration submode without committing any of the changes.

Step 5	(Optional) **show spanning-tree mst configuration** **Example:** `switch# show spanning-tree mst configuration`	Displays the MST configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to set the name of the MST region:

```
switch# config t
switch(config)#  spanning-tree mst configuration
switch(config-mst)# name accounting
switch(config-mst)# exit
switch(config)#
```

Specifying the MST Configuration Revision Number

You configure the revision number on the bridge. If two or more bridges are in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree mst configuration**
	Example:
	`switch(config)# spanning-tree mst configuration`
	`switch(config-mst)#`
		Enters MST configuration submode.

Step 3	**revision** *version* **Example:** `switch(config-mst)# revision 5`	Specifies the revision number for the MST region. The range is from 0 to 65535, and the default value is 0.

Step 4	**exit** or **abort**
	Example:
	`switch(config-mst)# exit`
	`switch(config)#`
		• **exit**
		Commits all the changes and exits MST configuration submode.
		• **abort**
		Exits the MST configuration submode without committing any of the changes.

Step 5	(Optional) **show spanning-tree mst configuration** **Example:** `switch# show spanning-tree mst configuration`	Displays the MST configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the revision number of the MSTI region to 5:

```
switch# config t
switch(config)# spanning-tree mst configuration
switch(config-mst)# revision 5
switch(config-mst)#
```

Configuring the Root Bridge

You can configure the device to become the MST root bridge.

The **spanning-tree vlan** *vlan_ID* **primary root** command fails if the value required to be the root bridge is less than 4096. If the software cannot lower the bridge priority any lower, the device returns the following message:

```
Error: Failed to set root bridge for VLAN 1
It may be possible to make the bridge root by setting the priority
for some (or all) of these instances to zero.
```

Note The root bridge for each MSTI should be a backbone or distribution device. Do not configure an access device as the spanning tree primary root bridge.

Enter the **diameter** keyword, which is available only for MSTI 0 (or the IST), to specify the Layer 2 network diameter (that is, the maximum number of Layer 2 hops between any two end stations in the Layer 2 network). When you specify the network diameter, the device automatically sets an optimal hello time, forward-delay time, and maximum-age time for a network of that diameter, which can significantly reduce the convergence time. You can enter the **hello** keyword to override the automatically calculated hello time.

Note With the device configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the **spanning-tree mst hello-timespanning-tree mst forward-time**, and **spanning-tree mst max-age** global configuration commands.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]] or **no spanning-tree mst** *instance-id* **root** **Example:**	• **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]] Configures a device as the root bridge as follows:

	Command or Action	Purpose
	`switch(config)# spanning-tree mst 5 root primary`	• For *instance-id*, specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094. • For **diameter** *net-diameter*, specify the maximum number of Layer 2 hops between any two end stations. The default is 7. This keyword is available only for MST instance 0. • For **hello-time** *seconds*, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is from 1 to 10 seconds; the default is 2 seconds. • **no spanning-tree mst** *instance-id* **root** Returns the switch priority, diameter, and hello time to default values.

Step 3	**exit** or **abort**
	Example:
	`switch(config)# exit`
	`switch#`
		• **exit**
		Commits all the changes and exits MST configuration submode.
		• **abort**
		Exits the MST configuration submode without committing any of the changes.

Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the device as the root switch for MSTI 5:

```
switch# config t
switch(config)# spanning-tree mst 5 root primary
switch(config)# exit
switch(config)#
```

Configuring an MST Secondary Root Bridge

You use this command on more than one device to configure multiple backup root bridges. Enter the same network diameter and hello-time values that you used when you configured the primary root bridge with the **spanning-tree mst root primary** global configuration command.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia*[**hello-time** *hello-time*]] or **no spanning-tree mst** *instance-id* **root** **Example:** `switch(config)# spanning-tree mst 0 root secondary`	- **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia*[**hello-time** *hello-time*]] Configures a device as the secondary root bridge as follows: - For *instance-id*, specify the single MSTI ID. - For **diameter** *net-diameter*, specify the maximum number of Layer 2 hops between any two end stations. The default is 7. This keyword is available only for MST instance 0. - For **hello-time** *seconds*, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is from 1 to 10 seconds; the default is 2 seconds. - **no spanning-tree mst** *instance-id* **root** Returns the switch priority, diameter, and hello-time to default values.

Step 3	**exit**
	Example:
	`switch# exit`
	`switch(config)#`
		Exits configuration mode.

	Command or Action	Purpose
Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the device as the secondary root switch for MSTI 0:

```
switch# config t
switch(config)# spanning-tree mst 0 root secondary
switch(config)# exit
switch#
```

Configuring the MST Switch Priority

You can configure the switch priority for an MST instance so that it is more likely that the specified device is chosen as the root bridge.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree mst** *instance-id* **priority** *priority-value* **Example:** `switch(config)# spanning-tree mst 0 priority 4096`	Configures a device priority as follows: • For *instance-id*, specify the single MSTI ID. • For *priority-value* the range is from 0 to 61440 in increments of 4096; the default is 32768. A lower number indicates that the device will most likely be chosen as the root bridge. Priority values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. The system rejects all other values.

	Command or Action	Purpose

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the priority of the bridge to 4096 for MSTI 0:

```
switch# config t
switch(config)# spanning-tree mst 0 priority 4096
switch(config)# exit
switch#
```

Configuring the MST Port Priority

If a loop occurs, MST uses the port priority when selecting an interface to put into the forwarding state. You can assign lower priority values to interfaces that you want selected first and higher priority values to the interface that you want selected last. If all interfaces have the same priority value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** {{*type slot/port*} | {**port-channel** *number*}}
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Specifies an interface to configure, and enters interface configuration mode.

	Command or Action	Purpose
Step 3	**spanning-tree mst** *instance-id* **port-priority** *priority* **Example:** `switch(config-if)# spanning-tree mst 0 port-priority 64`	Configures the port priority as follows: • For *instance-id*, specify the single MSTI ID. • For *priority*, the range is from 0 to 224 in increments of 32. The default is 128. A lower number indicates a higher priority. The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. The system rejects all other values.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface mode.

Step 5	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to set the MST interface port priority for MSTI 0 on Ethernet port 1/1 to 64:

```
switch# config t
switch(config)# interface ethernet 1/1
switch(config-if)# spanning-tree mst 0 port-priority 64
switch(config-if)# exit
switch(config)#
```

Configuring the MST Port Cost

The MST port cost default value is derived from the media speed of an interface. If a loop occurs, MST uses the cost when selecting an interface to put in the forwarding state. You can assign lower cost values to interfaces that you want selected first and higher cost to interfaces values that you want selected last. If all interfaces have the same cost value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

Note MST uses the long path-cost calculation method.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** {{*type slot/port*} | {**port-channel** *number*}}
	Example:
	`switch# config t`
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**spanning-tree mst** *instance-id* **cost** {*cost* | *auto*}
	Example:
	`switch(config-if)# spanning-tree mst 0 cost 17031970`
		Configures the cost.
		If a loop occurs, MST uses the path cost when selecting an interface to place into the forwarding state. A lower path cost represents higher-speed transmission as follows:
		• For *instance-id*, specify the single MSTI ID.
		• For *cost*, the range is from 1 to 200000000. The default value is **auto**, which is derived from the media speed of the interface.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface mode.

Step 5	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to set the MST interface port cost on Ethernet 1/1 for MSTI 0:

```
switch# config t
switch(config)# interface ethernet 1/1
switch(config-if)# spanning-tree mst 0 cost 17031970
switch(config-if)# exit
switch(config)#
```

Configuring the MST Hello Time

You can configure the interval between the generation of configuration messages by the root bridge for all instances on the device by changing the hello time.

Note Be careful when using the **spanning-tree mst hello-time** command. For most situations, we recommend that you enter the **spanning-tree mst** *instance-id* **root primary** and the **spanning-tree mst** *instance-id* **root secondary** global configuration commands to modify the hello time.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree mst hello-time** *seconds* **Example:** `switch(config)# spanning-tree mst hello-time 1`	Configures the hello time for the MST instance. The hello time is the interval between the generation of configuration messages by the root bridge. These messages mean that the device is alive. For *seconds*, the range is from 1 to 10, and the default is 2 seconds.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config)# copy running-config startup-config`	

Example

This example shows how to configure the hello time of the device to 1 second:

```
switch# config t
switch(config)# spanning-tree mst hello-time 1
switch(config)# exit
switch#
```

Configuring the MST Forwarding-Delay Time

You can set the forward delay timer for the MST instance on the device with one command.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree mst forward-time** *seconds* **Example:** `switch(config)# spanning-tree mst forward-time 10`	Configures the forward time for the MST instance. The forward delay is the number of seconds that a port waits before changing from its spanning tree blocking and learning states to the forwarding state. For *seconds*, the range is from 4 to 30, and the default is 15 seconds.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the forward-delay time of the device to 10 seconds:

```
switch# config t
switch(config)# spanning-time mst forward-time 10
switch(config)# exit
switch#
```

Configuring the MST Maximum-Aging Time

You can set the maximum-aging timer for the MST instance on the device with one command (the maximum age time only applies to the IST).

The maximum-aging timer is the number of seconds that a device waits without receiving spanning tree configuration messages before attempting a reconfiguration.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree mst max-age** *seconds* **Example:** `switch(config)# spanning-tree mst max-age 40`	Configures the maximum-aging time for the MST instance. The maximum-aging time is the number of seconds that a device waits without receiving spanning tree configuration messages before attempting a reconfiguration. For *seconds*, the range is from 6 to 40, and the default is 20 seconds.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the maximum-aging timer of the device to 40 seconds:

```
switch# config t
switch(config)# spanning-tree mst max-age 40
switch(config)# exit
switch#
```

Configuring the MST Maximum-Hop Count

You can configure the maximum hops inside the region and apply it to the IST and the MST instance in that region. MST uses the path cost to the IST regional root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism. The hop count achieves the same result as the message-age information (triggers a reconfiguration).

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree mst max-hops** *hop-count* **Example:** `switch(config)# spanning-tree mst max-hops 40`	Specifies the number of hops in a region before the BPDU is discarded and the information held for a port is aged. For *hop-count*, the range is from 1 to 255, and the default value is 20 hops.

Step 3	**exit**
	Example:
	`switch(config-mst)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to set the maximum hops to 40:

```
switch# config t
switch(config)# spanning-tree mst max-hops 40
switch(config)# exit
switch#
```

Configuring an Interface to Proactively Send Prestandard MSTP Messages - CLI Version

By default, interfaces on a device running MST send prestandard, rather than standard, MSTP messages after they receive a prestandard MSTP message from another interface. You can configure the interface to proactively send prestandard MSTP messages. That is, the specified interface would not have to wait to receive a prestandard MSTP message; the interface with this configuration always sends prestandard MSTP messages.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure and enters the interface configuration mode.

Step 3	**spanning-tree mst pre-standard** **Example:** `switch(config-if)# spanning-tree mst pre-standard`	Specifies that the interface always sends MSTP messages in the prestandard format, rather than in the MSTP standard format.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface mode.

Step 5	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to set the MST interface so that it always sends MSTP messages in the prestandard format:

```
switch# config t
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree mst pre-standard
switch(config-if)# exit
switch(config)#
```

Specifying the Link Type for MST - CLI Version

Rapid connectivity (802.1w standard) is established only on point-to-point links. By default, the link type is controlled from the duplex mode of the interface. A full-duplex port is considered to have a point-to-point connection; a half-duplex port is considered to have a shared connection.

If you have a half-duplex link physically connected point to point to a single port on a remote device, you can override the default setting on the link type and enable rapid transitions.

If you set the link to shared, STP falls back to 802.1D.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure and enters the interface configuration mode.

Step 3	**spanning-tree link-type** {*auto* | *point-to-point* | *shared*}
	Example:
	`switch(config-if)# spanning-tree link-type point-to-point`
		Configures the link type to be either a point-to-point link or shared link. The system reads the default value from the device connection, as follows: half duplex links are shared and full-duplex links are point to point. If the link type is shared, the STP falls back to 802.1D. The default is auto, which sets the link type based on the duplex setting of the interface.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface mode.

	Command or Action	Purpose
Step 5	(Optional) **show spanning-tree** **Example:** `switch# show spanning-tree`	Displays the STP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the link type as a point-to-point link:

```
switch# config t
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree link-type point-to-point
switch(config-if)# exit
switch(config)#
```

Reinitializing the Protocol for MST

An MST bridge can detect that a port is at the boundary of a region when it receives a legacy BPDU or an MST BPDU that is associated with a different region. However, the STP protocol migration cannot determine whether the legacy device, which is a device that runs only IEEE 802.1D, has been removed from the link unless the legacy device is the designated switch. Enter this command to reinitialize the protocol negotiation (force the renegotiation with neighboring devices) on the entire device or on specified interfaces.

Procedure

	Command or Action	Purpose
Step 1	**clear spanning-tree detected-protocol** [**interface** *interface* [*interface-num* | *port-channel*]] **Example:** `switch# clear spanning-tree detected-protocol`	Reintializes MST on an entire device or specified interfaces.

Example

This example shows how to reinitialize MST on the Ethernet interface on slot 1, port 8:

```
switch# clear spanning-tree detected-protocol interface ethernet 1/8
```

Verifying the MST Configuration

To display MST configuration information, perform one of the following tasks:

Command	Purpose
show running-config spanning-tree [**all**]	Displays STP information.
show spanning-tree mst configuration	Displays MST information.
show spanning-tree mst [**detail**]	Displays information about MST instances.
show spanning-tree mst *instance-id* [**detail**]	Displays information about the specified MST instance.
show spanning-tree mst *instance-id* **interface** {**ethernet** *slot/port* | **port-channel** *channel-number*} [**detail**]	Displays MST information for the specified interface and instance.
show spanning-tree summary	Displays summary STP information.
show spanning-tree detail	Displays detailed STP information.
show spanning-tree {**vlan** *vlan-id* | **interface** {[**ethernet** *slot/port*] | [**port-channel** *channel-number*]}} [**detail**]	Displays STP information per VLAN and interface.
show spanning-tree vlan *vlan-id* **bridge**	Displays information on the STP bridge.

Displaying and Clearing MST Statistics -- CLI Version

To display MST configuration information, perform one of the following tasks:

Command	Purpose
clear spanning-tree counters [**interface** *type slot/port* | **vlan***vlan-id*]	Clears the counters for STP.
show spanning-tree {**vlan** *vlan-id* | **interface** {[**ethernet** *slot/port*] | [**port-channel***channel-number*]}} **detail**	Displays information about STP by interface or VLAN including BPDUs sent and received.

MST Example Configuration

The following example shows how to configure MST:

```
switch# configure terminal
switch(config)# spanning-tree mode mst
switch(config)# spanning-tree port type edge bpduguard default
switch(config)# spanning-tree port type edge bpdufilter default
switch(config)# spanning-tree port type network default
switch(config)# spanning-tree mst 0 priority 24576
switch(config)# spanning-tree mst configuration
switch(config-mst)# name cisco_region_1
switch(config-mst)# revision 2
switch(config-mst)# instance 1 vlan 1-21
```

Additional References for MST -- CLI Version

Related Documents

Related Topic	Document Title
Layer 2 interfaces	*Cisco Nexus® 3550-T Interfaces Configuration* section
NX-OS fundamentals	*Cisco Nexus Series NX-OS Fundamentals Configuration Guide*
High availability	*Cisco Nexus Series High Availability and Redundancy Guide*
System management	*Cisco Nexus® 3550-T System Management Configuration* section

Standards

Standards	Title
IEEE 802.1Q-2006 (formerly known as IEEE 802.1s), IEEE 802.1D-2004 (formerly known as IEEE 802.1w), IEEE 802.1D, IEEE 802.1t	—

MIBS

MIBs	MIBs Link
CISCO-STP-EXTENSION-MIB BRIDGE-MIB	To locate and download MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.htm

CHAPTER 27

Configuring STP Extensions Using Cisco NX-OS

Information About STP Extensions

Note See the *Cisco Nexus® 3550-T Interfaces Configuration Guide*, for information on creating Layer 2 interfaces.

Cisco has added extensions to STP that enhances loop prevention, protects against some possible user misconfigurations, and provides better control over the protocol parameters. Although, in some cases, similar functionality may be incorporated into the IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) standard, we recommend using these extensions. All of these extensions, except PVST Simulation, can be used with MST. You use PVST Simulation only with MST.

The available extensions are spanning tree edge ports (which supply the functionality previously known as PortFast), Bridge Assurance, BPDU Guard, BPDU Filtering, Loop Guard, Root Guard, and PVT Simulation. Many of these features can be applied either globally or on specified interfaces.

Note Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the text is discussing the IEEE 802.1D Spanning Tree Protocol, 802.1D is stated specifically.

STP Port Types

You can configure a spanning tree port as an edge port, a network port, or a normal port. A port can be in only one of these states at a given time. The default spanning tree port type is normal.

Edge ports, which are connected to Layer 2 hosts, can be either an access port or a trunk port.

Note If you configure a port connected to a Layer 2 switch or bridge as an edge port, you might create a bridging loop.

STP Edge Ports

You connect STP edge ports only to Layer 2 hosts. The edge port interface immediately transitions to the forwarding state, without moving through the blocking or learning states. (This immediate transition was previously configured as the Cisco-proprietary feature PortFast.)

Interfaces that are connected to Layer 2 hosts should not receive STP bridge protocol data units (BPDUs).

BPDU Guard

Enabling BPDU Guard shuts down that interface if a BPDU is received.

You can configure BPDU Guard at the interface level. When configured at the interface level, BPDU Guard shuts the port down as soon as the port receives a BPDU, regardless of the port type configuration.

When you configure BPDU Guard globally, it is effective only on operational spanning tree edge ports. In a valid configuration, Layer 2 LAN edge interfaces do not receive BPDUs. A BPDU that is received by an edge Layer 2 LAN interface signals an invalid configuration, such as the connection of an unauthorized device. BPDU Guard, when enabled globally, shuts down all spanning tree edge ports when they receive a BPDU.

BPDU Guard provides a secure response to invalid configurations, because you must manually put the Layer 2 LAN interface back in service after an invalid configuration.

Note When enabled globally, BPDU Guard applies to all operational spanning tree edge interfaces.

BPDU Filtering

You can use BPDU Filtering to prevent the device from sending or even receiving BPDUs on specified ports.

When configured globally, BPDU Filtering applies to all operational spanning tree edge ports. You should connect edge ports only to hosts, which typically drop BPDUs. If an operational spanning tree edge port receives a BPDU, it immediately returns to a normal spanning tree port type and moves through the regular transitions. In that case, BPDU Filtering is disabled on this port, and spanning tree resumes sending BPDUs on this port.

In addition, you can configure BPDU Filtering by the individual interface. When you explicitly configure BPDU Filtering on a port, that port does not send any BPDUs and drops all BPDUs that it receives. You can effectively override the global BPDU Filtering setting on individual ports by configuring the specific interface. This BPDU Filtering command on the interface applies to the entire interface, whether the interface is trunking or not.

Caution Use care when configuring BPDU Filtering per interface. If you explicitly configure BPDU Filtering on a port that is not connected to a host, it can result in bridging loops because the port will ignore any BPDU that it receives and go to forwarding.

This table lists all the BPDU Filtering combinations.

Table 24: BPDU Filtering Configurations

BPDU Filtering Per Port Configuration	BPDU Filtering Global Configuration	STP Edge Port Configuration	BPDU Filtering State
Default [1]	Enable	Enable	Enable [2]
Default	Enable	Disable	Disable
Default	Disable	Not applicable	Disable
Disable	Not applicable	Not applicable	Disable
Enable	Not applicable	Not applicable	Enable

[1] No explicit port configuration.

[2] The port transmits at least 10 BPDUs. If this port receives any BPDUs, the port returns to the spanning tree normal port state and BPDU filtering is disabled.

Loop Guard

Loop Guard helps prevent bridging loops that could occur because of a unidirectional link failure on a point-to-point link.

An STP loop occurs when a blocking port in a redundant topology erroneously transitions to the forwarding state. Transitions are usually caused by a port in a physically redundant topology (not necessarily the blocking port) that stops receiving BPDUs.

When you enable Loop Guard globally, it is useful only in switched networks where devices are connected by point-to-point links. On a point-to-point link, a designated bridge cannot disappear unless it sends an inferior BPDU or brings the link down. However, you can enable Loop Guard on shared links per interface,

You can use Loop Guard to determine if a root port or an alternate/backup root port receives BPDUs. If the port that was previously receiving BPDUs is no longer receiving BPDUs, Loop Guard puts the port into an inconsistent state (blocking) until the port starts to receive BPDUs again. If such a port receives BPDUs again, the port—and link—is deemed viable again. The protocol removes the loop-inconsistent condition from the port, and the STP determines the port state because the recovery is automatic.

Loop Guard isolates the failure and allows STP to converge to a stable topology without the failed link or bridge. Disabling Loop Guard moves all loop-inconsistent ports to the listening state.

You can enable Loop Guard on a per-port basis. When you enable Loop Guard on a port, it is automatically applied to all of the active instances or VLANs to which that port belongs. When you disable Loop Guard, it is disabled for the specified ports.

Enabling Loop Guard on a root device has no effect but provides protection when a root device becomes a nonroot device.

Root Guard

When you enable Root Guard on a port, Root Guard does not allow that port to become a root port. If a received BPDU triggers an STP convergence that makes that designated port become a root port, that port is put into a root-inconsistent (blocked) state. After the port stops receiving superior BPDUs, the port is unblocked again. Through STP, the port moves to the forwarding state. Recovery is automatic.

When you enable Root Guard on an interface, this functionality applies to all VLANs to which that interface belongs.

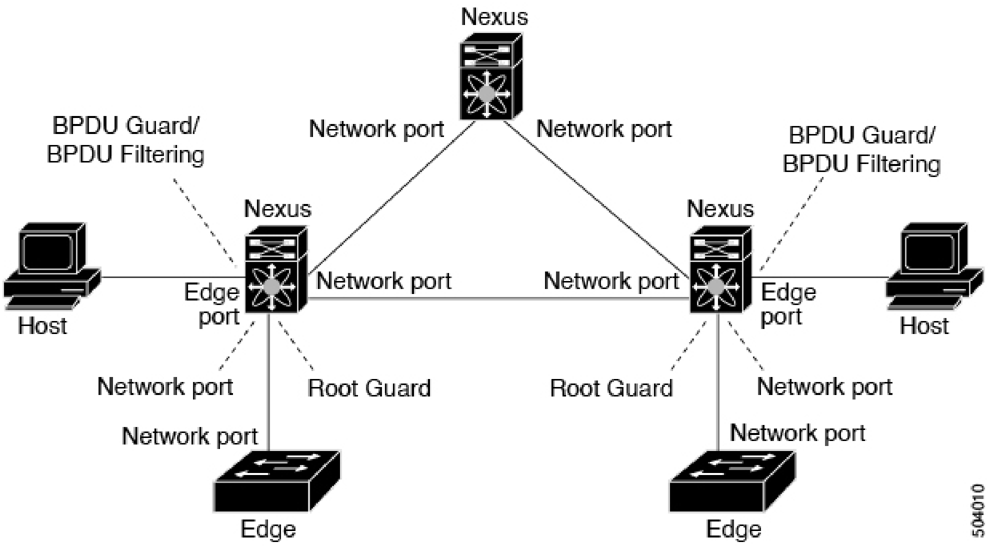
You can use Root Guard to enforce the root bridge placement in the network. Root Guard ensures that the port on which Root Guard is enabled is the designated port. Normally, root bridge ports are all designated ports, unless two or more of the ports of the root bridge are connected. If the bridge receives superior BPDUs on a Root Guard-enabled port, the bridge moves this port to a root-inconsistent STP state. In this way, Root Guard enforces the position of the root bridge.

You cannot configure Root Guard globally.

Applying STP Extension Features

Figure 24: Network with STP Extensions Correctly Deployed

We recommend that you configure the various STP extension features through your network as shown in this figure. Bridge Assurance is enabled on the entire network. You should enable either BPDU Guard or BPDU Filtering on the host interface.



PVST Simulation

MST operates with no need for user configuration. The PVST simulation feature enables this interoperability.

Note PVST simulation is enabled by default when you enable MST. By default, all interfaces on the device operate on MST.

The root bridge for all STP instances must all be in the MST region. If the root bridge for all STP instances are not on MST, the software moves the port into a PVST simulation-inconsistent state.

Note We recommend that you put the root bridge for STP instances in the MST region. Only default STP instance is supported in Cisco Nexus® 3550-T.

High Availability for STP

The software supports high availability for STP. However, the statistics and timers are not restored when STP restarts. The timers start again and the statistics begin from 0.

Note See the *Cisco Nexus Series NX-OS High Availability and Redundancy Guide*, for complete information on high-availability features.

Prerequisites for STP Extensions

STP has the following prerequisites:

- You must be logged onto the device.
- You must have STP configured already.

Guidelines and Limitations for Configuring STP Extensions

STP extensions have the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- Connect STP network ports only to switches.
- You should configure host ports as STP edge ports and not as network ports.
- You should configure all access and trunk ports connected to Layer 2 hosts as edge ports.
- We recommend that you enable BPDU Guard on all edge ports.
- Enabling Loop Guard globally works only on point-to-point links.
- Enabling Loop Guard per interface works on both shared and point-to-point links.
- Root Guard forces a port to always be a designated port; it does not allow a port to become a root port. Loop Guard is effective only if the port is a root port or an alternate port. You cannot enable Loop Guard and Root Guard on a port at the same time.
- Loop Guard has no effect on a disabled spanning tree instance or a VLAN.

- Spanning tree always chooses the first operational port in the channel to send the BPDUs. If that link becomes unidirectional, Loop Guard blocks the channel, even if other links in the channel are functioning properly.
- If you group together a set of ports that are already blocked by Loop Guard to form a channel, spanning tree loses all the state information for those ports and the new channel port may obtain the forwarding state with a designated role.
- If a channel is blocked by Loop Guard and the channel members go back to an individual link status, spanning tree loses all the state information. The individual physical ports may obtain the forwarding state with the designated role, even if one or more of the links that formed the channel are unidirectional.
- You should enable Loop Guard globally on a switch network with physical loops.
- You should enable Root Guard on ports that connect to network devices that are not under direct administrative control.

Default Settings for STP Extensions

This table lists the default settings for STP extensions.

Table 25: Default STP Extension Parameters

Parameters	Default
Port type	Normal
Global BPDU Guard	Disabled
BPDU Guard per interface	Disabled
Global BPDU Filtering	Disabled
BPDU Filtering per interface	Disabled
Global Loop Guard	Disabled
Loop Guard per interface	Disabled
Root Guard per interface	Disabled

Configuring STP Extensions Steps

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

You can enable Loop Guard per interface on either shared or point-to-point links.

Configuring Spanning Tree Port Types Globally

The spanning tree port type designation depends on the device the port is connected to, as follows:

- Edge—Edge ports are connected to Layer 2 hosts and are access ports.
- Normal—Normal ports are neither edge ports nor network ports; they are normal spanning tree ports. These ports can be connected to any device.

You can configure the port type either globally or per interface. By default, the spanning tree port type is normal.

Before you begin

Before you configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring the ports correctly as to the device to which the port is connected.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree port type edge default** or **spanning-tree port type network default** **Example:** `switch(config)# spanning-tree port type edge default`	• **spanning-tree port type edge default** Configures all access ports connected to Layer 2 hosts as edge ports. Edge ports immediately transition to the forwarding state without passing through the blocking or learning state at linkup. By default, spanning tree ports are normal port types. • **spanning-tree port type network default** Configures all interfaces connected to Layer 2 switches and bridges as spanning tree network ports. If you enable Bridge Assurance, it automatically runs on network ports. By default, spanning tree ports are normal port types. **Note** If you configure interfaces connected to Layer 2 hosts as network ports, those ports automatically move into the blocking state.

	Command or Action	Purpose

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	Displays the STP configuration including STP port types if configured.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure all access ports connected to Layer 2 hosts as spanning tree edge ports:

```
switch# config t
switch(config)# spanning-tree port type edge default
switch(config)# exit
switch#
```

Configuring Spanning Tree Edge Ports on Specified Interfaces

You can configure spanning tree edge ports on specified interfaces. Interfaces configured as spanning tree edge ports immediately transition to the forwarding state, without passing through the blocking or learning states, on linkup.

This command has four states:

- **spanning-tree port type edge**—This command explicitly enables edge behavior on the access port.
- **spanning-tree port type edge trunk**—This command explicitly enables edge behavior on the trunk port.

Note If you enter the **spanning-tree port type edge trunk** command, the port is configured as an edge port even in the access mode.

- **spanning-tree port type normal**—This command explicitly configures the port as a normal spanning tree port and the immediate transition to the forwarding state is not enabled.
- **no spanning-tree port type**—This command implicitly enables edge behavior if you define the **spanning-tree port type edge default** command in global configuration mode. If you do not configure

the edge ports globally, the **no spanning-tree port type** command is equivalent to the **spanning-tree port type normal** command.

Before you begin

Before you configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring the ports correctly as to the device to which the port is connected.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 3	**spanning-tree port type edge** **Example:** `switch(config-if)# spanning-tree port type edge`	Configures the specified access interfaces to be spanning edge ports. Edge ports immediately transition to the forwarding state without passing through the blocking or learning state at linkup. By default, spanning tree ports are normal port types.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 5	(Optional) **show spanning-tree interface** *type slot/port* **ethernet** *x/y* **Example:** `switch# show spanning-tree ethernet 1/4`	Displays the STP configuration including the STP port type if configured.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the Ethernet access interface 1/4 to be a spanning tree edge port:

```
switch# config t
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port type edge
switch(config-if)# exit
switch(config)#
```

Enabling BPDU Guard Globally

You can enable BPDU Guard globally by default. In this condition, the system shuts down an edge port that receives a BPDU.

Note We recommend that you enable BPDU Guard on all edge ports.

Before you begin

Before you configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring the ports correctly as to the device to which the port is connected.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree port type edge bpduguard default** **Example:** `switch(config)# spanning-tree port type edge bpduguard default`	Enables BPDU Guard by default on all spanning tree edge ports. By default, global BPDU Guard is disabled.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

	Command or Action	Purpose
Step 4	(Optional) **show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	Displays summary STP information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enable BPDU Guard on all spanning tree edge ports:

```
switch# config t
switch(config)# spanning-tree port type edge bpduguard default
switch(config)# exit
switch#
```

Enabling BPDU Guard on Specified Interfaces

You can enable BPDU Guard on specified interfaces. Enabling BPDU Guard shuts down the port if it receives a BPDU.

You can configure BPDU Guard on specified interfaces as follows:

- **spanning-tree bpduguard enable** —Unconditionally enables BPDU Guard on the interface.
- **spanning-tree bpduguard disable** —Unconditionally disables BPDU Guard on the interface.
- **no spanning-tree bpduguard** —Enables BPDU Guard on the interface if it is an operational edge port and if the **spanning-tree port type edge bpduguard default** command is configured.

Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 3	**spanning-tree bpduguard** {**enable** | **disable**} or **no spanning-tree bpduguard** **Example:** `switch(config-if)# spanning-tree bpduguard enable`	- **spanning-tree bpduguard** {**enable** | **disable**} Enables or disables BPDU Guard for the specified spanning tree edge interface. By default, BPDU Guard is disabled on the interfaces. - **no spanning-tree bpduguard** Falls back to the default BPDU Guard global setting that you set for the interfaces by entering the **spanning-tree port type edge bpduguard default** command.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface mode.

Step 5	(Optional) **show spanning-tree interface** *type slot/port* **detail** **Example:** `switch# show spanning-tree interface ethernet detail`	Displays summary STP information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to explicitly enable BPDU Guard on the Ethernet edge port 1/4:

```
switch# config t
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree bpduguard enable
switch(config-if)# exit
switch(config)#
```

Enabling BPDU Filtering Globally

You can enable BPDU Filtering globally by default on spanning tree edge ports.

If an edge port with BPDU Filtering enabled receives a BPDU, it loses its operation status as edge port and resumes the regular STP transitions. However, this port maintains its configuration as an edge port.

Caution Be careful when using this command. Using this command incorrectly can cause bridging loops.

Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.
- Ensure that you have configured some spanning tree edge ports.

Note When enabled globally, BPDU Filtering is applied only on ports that are operational edge ports. Ports send a few BPDUs at linkup before they effectively filter outbound BPDUs. If a BPDU is received on an edge port, it immediately loses its operational edge port status and BPDU Filtering is disabled.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree port type edge bpdufilter default** **Example:** `switch(config)# spanning-tree port type edge bpdufilter default`	Enables BPDU Filtering by default on all operational spanning tree edge ports. Global BPDU Filtering is disabled by default.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	Displays summary STP information.

	Command or Action	Purpose
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enable BPDU Filtering on all operational spanning tree edge ports:

```
switch# config t
switch(config)# spanning-tree port type edge bpdufilter default
switch(config)# exit
switch#
```

Enabling BPDU Filtering on Specified Interfaces

You can apply BPDU Filtering to specified interfaces. When enabled on an interface, that interface does not send any BPDUs and drops all BPDUs that it receives. This BPDU Filtering functionality applies to the entire interface, whether trunking or not.

Caution Be careful when you enter the **spanning-tree bpdufilter enable** command on specified interfaces. Explicitly configuring BPDU Filtering on a port that is not connected to a host can result in bridging loops because the port will ignore any BPDU that it receives and go to forwarding.

You can enter this command to override the port configuration on specified interfaces.

This command has three states:

- **spanning-tree bpdufilter enable**—Unconditionally enables BPDU Filtering on the interface.
- **spanning-tree bpdufilter disable**—Unconditionally disables BPDU Filtering on the interface.
- **no spanning-tree bpdufilter** ——Enables BPDU Filtering on the interface if the interface is in operational edge port and if you configure the **spanning-tree port type edge bpdufilter default** command.

Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.

Note When you enable BPDU Filtering locally on a port, this feature prevents the device from receiving or sending BPDUs on this port.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 3	**spanning-tree bpdufilter** {**enable** | **disable**} or **no spanning-tree bpdufilter** **Example:** `switch(config-if)# spanning-tree bpdufilter enable`	• **spanning-tree bpdufilter** {**enable** | **disable**} Enables or disables BPDU Filtering for the specified spanning tree edge interface. By default, BPDU Filtering is disabled. • **no spanning-tree bpdufilter** Enables BPDU Filtering on the interface if the interface is an operational spanning tree edge port and if you enter the **spanning-tree port type edge bpdufilter default** command.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface mode.

Step 5	(Optional) **show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	Displays summary STP information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to explicitly enable BPDU Filtering on the Ethernet spanning tree edge port 1/4:

```
switch# config t
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree bpdufilter enable
switch(config-if)# exit
switch(config)#
```

Enabling Loop Guard Globally

You can enable Loop Guard globally by default on all point-to-point spanning tree normal and network ports. Loop Guard does not run on edge ports.

Loop Guard provides additional security in the bridge network. Loop Guard prevents alternate or root ports from becoming the designated port because of a failure that could lead to a unidirectional link.

Note Entering the Loop Guard command for the specified interface overrides the global Loop Guard command.

Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.
- Ensure that you have spanning tree normal ports or have configured some network ports.

Procedure

	Command or Action	Purpose

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

Step 2	**spanning-tree loopguard default** **Example:** `switch(config)# spanning-tree loopguard default`	Enables Loop Guard by default on all spanning tree normal and network ports. By default, global Loop Guard is disabled.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	Displays summary STP information.

	Command or Action	Purpose
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enable Loop Guard on all spanning tree normal or network ports:

```
switch# config t
switch(config)# spanning-tree loopguard default
switch(config)# exit
switch#
```

Enabling Loop Guard or Root Guard on Specified Interfaces

Note You can run Loop Guard on spanning tree normal or network ports. You can run Root Guard on all spanning tree ports: normal, edge, or network.

You can enable either Loop Guard or Root Guard on specified interfaces.

Enabling Root Guard on a port means that port cannot become a root port, and Loop Guard prevents alternate or root ports from becoming the designated port because of a failure that could lead to a unidirectional link.

Both Loop Guard and Root Guard enabled on an interface apply to all VLANs to which that interface belongs.

Note Entering the Loop Guard command for the specified interface overrides the global Loop Guard command.

Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring Loop Guard on spanning tree normal or network ports.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 3	**spanning-tree guard** {**loop** | **root** | **none**}
	Example:
	`switch(config-if)# spanning-tree guard loop`
		Enables or disables either Loop Guard or Root Guard for the specified interface. By default, Root Guard is disabled by default, and Loop Guard on specified ports is also disabled.
		Note Loop Guard runs only on spanning tree normal and network interfaces. This example shows Loop Guard is enabled on the specified interface.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface mode.

Step 5	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/10`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 6	**spanning-tree guard** {**loop** | **root** | **none**}
	Example:
	`switch(config-if)# spanning-tree guard root`
		Enables or disables either Loop Guard or Root Guard for the specified interface. By default, Root Guard is disabled by default, and Loop Guard on specified ports is also disabled.
		The example shows Root Guard is enabled on a different interface.

Step 7	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface mode.

Step 8	(Optional) **show spanning-tree interface** *type slot/port* **detail** **Example:** `switch# show spanning-tree interface ethernet 1/4 detail`	Displays summary STP information.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enable Root Guard on Ethernet port 1/4:

```
switch# config t
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree guard root
switch(config-if)# exit
switch(config)#
```

Verifying the STP Extension Configuration

To display the configuration information for the STP extensions, perform one of the following tasks:

Command	Purpose
show running-config spanning-tree [**all**]	Displays information about STP.
show spanning-tree summary	Displays summary information on STP.
show spanning-tree mst *instance-id* **interface** {**ethernet** *slot/port* | **port-channel** *channel-number*} [**detail**]	Displays MST information for the specified interface and instance.

Configuration Examples for STP Extension

The following example shows how to configure the STP extensions:

```
switch# configure terminal
switch(config)# spanning-tree port type network default
switch(config)# spanning-tree port type edge bpduguard default
switch(config)# spanning-tree port type edge bpdufilter default

switch(config)# interface ethernet 1/1
switch(config-if)# spanning-tree port type edge
switch(config-if)# exit

switch(config)# interface ethernet 1/2
switch(config-if)# spanning-tree port type edge
switch(config-if)# exit
switch(config)#
```

Additional References for STP Extensions -- CLI Version

Related Documents

Related Topic	Document Title
Layer 2 interfaces	*Cisco Nexus® 3550-T Interfaces Configuration* section
NX-OS fundamentals	*Cisco Nexus® 3550-T Fundamentals Configuration* section

Related Topic	Document Title
System management	*Cisco Nexus® 3550-T System Management Configuration Guide* section
	Cisco NX-OS Licensing Guide

Standards

Standards	Title
IEEE 802.1Q-2006 (formerly known as IEEE 802.1s), IEEE 802.1D-2004 (formerly known as IEEE 802.1w), IEEE 802.1D, IEEE 802.1t	—

MIBs

MIBs	MIBs Link
• CISCO-STP-EXTENSION-MIB • BRIDGE-MIB	To locate and download MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

PART VII

Cisco Nexus 3550-T Interfaces Configuration Guide

CHAPTER 28

Interfaces Configuration Guide

This preface includes the following sections:

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

About Interfaces

Cisco NX-OS supports multiple configuration parameters for each of the interface types supported. Most of these parameters are covered in this guide but some are described in other documents.

The following table shows where to get further information on the parameters you can configure for an interface.

Table 26: Interface Parameters

Feature	Parameters	Further Information
Basic parameters	Description, duplex, error disable, flow control, beacon	*Configuring Basic Interface Parameters*
Layer 3	Medium, IPv4 addresses	*Configuring Layer 3 Interfaces*
Layer 3	Bandwidth, delay, IP routing, VRFs	*Cisco Nexus® 3550-T Unicast Routing Configuration* section *Cisco Nexus® 3550-T Multicast Routing Configuration* section
Port Channels	Channel group, LACP	*Configuring Port Channels*

Feature	Parameters	Further Information
Security	EOU	*Cisco Nexus® 3550-T Security Configuration* section

Ethernet Interfaces

Ethernet interfaces include routed ports.

Cisco Nexus® 3550-T switch has the following guidelines and limitations:

- Cisco Nexus® 3550-T supports only 10G speed.

Access Ports

An access port carries traffic for one VLAN. This type of port is a Layer 2 interface only.

For more information on access ports, see the "Information About Access and Trunk Interfaces" section.

Trunk Ports

A trunk port transmits untagged packets for one VLAN plus encapsulated, tagged, packetsfor multiple VLANs. (See the IEEE 802.1Q Encapsulation section for information about encapsulation.)

You can configure Layer 2 switching ports as access or trunk ports. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network. All Layer 2 switching ports maintain MAC address tables.

Routed Ports

A routed port is a physical port that can route IP traffic to another device. A routed port is a Layer 3 interface only.

For more information on routed ports, see the *Routed Interfaces* section.

Management Interface

You can use the management Ethernet interface to connect the device to a network for remote management using a Telnet client, the Simple Network Management Protocol (SNMP), or other management agents. The management port (mgmt0) is autosensing and operates in full-duplex mode at a speed of 10/100/1000 Mb/s.

For more information on the management interface, see the Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide. You will also find information on configuring the IP address and default IP routing for the management interface in this document.

Port-Channel Interfaces

A port channel is a logical interface that is an aggregation of multiple physical interfaces. You can bundle up to 4 individual links to physical ports into a port channel to improve bandwidth and redundancy. You can also use port channeling to load balance traffic across these channeled physical interfaces. For more information about port-channel interfaces, see the *Configuring Port Channels* section.

Loopback Interfaces

A virtual loopback interface is a virtual interface with a single endpoint that is always up. Any packet that is transmitted over a virtual loopback interface is immediately received by that interface. Loopback interfaces emulate a physical interface.

High Availability for Interfaces

Interfaces support stateful and stateless restarts.

CHAPTER 29

Configuring Static NAT Translation

Network Address Translation Overview

Network Address Translation (NAT) enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT operates on a device, usually connecting two networks, and translates private (not globally unique) IP addresses in the internal network into legal IP addresses before packets are forwarded to another network. You can configure NAT to advertise only one IP address for the entire network to the outside world. This ability provides additional security, effectively hiding the entire internal network behind one IP address.

A device configured with NAT has at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit router between a stub domain and a backbone. When a packet leaves the domain, NAT translates the locally significant source IP address into a globally unique IP address. When a packet enters the domain, NAT translates the globally unique destination IP address into a local IP address. If more than one exit point exists, NAT configured at each point must have the same translation table.

NAT is described in RFC 1631.

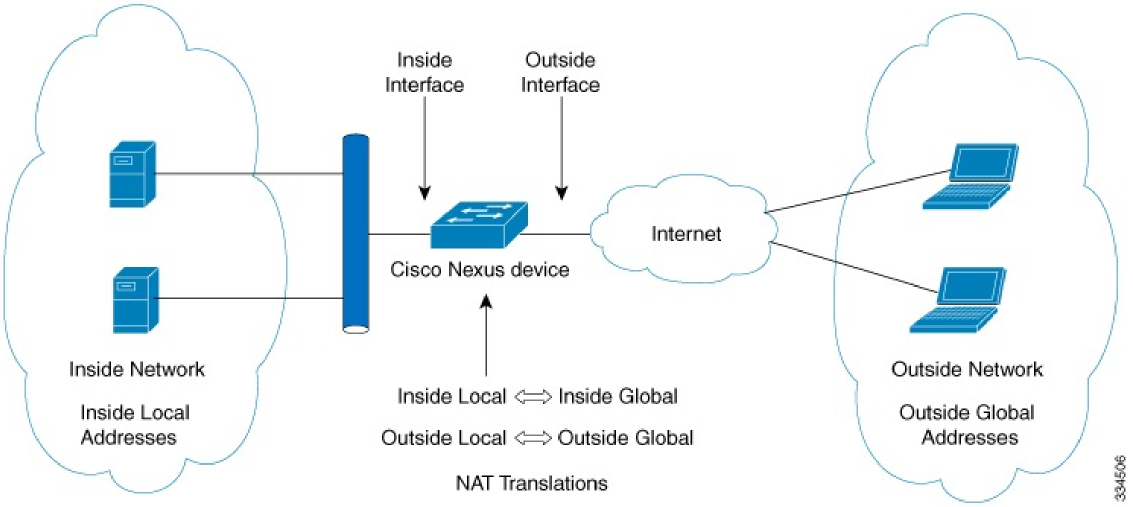
Information About Static NAT

Static Network Address Translation (NAT) allows the user to configure one-to-one translations of the inside local addresses to the outside global addresses. It allows both IP addresses and port number translations from the inside to the outside traffic and the outside to the inside traffic. The Cisco Nexus® device supports Hitless NAT, which means that you can add or remove a NAT translation in the NAT configuration without affecting the existing NAT traffic flows.

Static NAT creates a fixed translation of private addresses to public addresses. Because static NAT assigns addresses on a one-to-one basis, you need an equal number of public addresses as private addresses. Because the public address is the same for each consecutive connection with static NAT, and a persistent translation rule exists, static NAT enables hosts on the destination network to initiate traffic to a translated host if an access list exists that allows it .

The figure shows a typical static NAT scenario. The translation is always active so both translated and remote hosts can originate connections, and the mapped address is statically assigned by the **static** command.

Figure 25: Static NAT



These are key terms to help you understand static NAT:

- NAT inside interface—The Layer 3 interface that faces the private network.
- NAT outside interface—The Layer 3 interface that faces the public network.
- Local address—Any address that appears on the inside (private) portion of the network.
- Global address—Any address that appears on the outside (public) portion of the network.
- Legitimate IP address—An address that is assigned by the Network Information Center (NIC) or service provider.
- Inside local address—The IP address assigned to a host on the inside network. This address does not need to be a legitimate IP address.
- Outside local address—The IP address of an outside host as it appears to the inside network. It does not have to be a legitimate address, because it is allocated from an address space that can be routed on the inside network.
- Inside global address—A legitimate IP address that represents one or more inside local IP addresses to the outside world.
- Outside global address—The IP address that the host owner assigns to a host on the outside network. The address is a legitimate address that is allocated from an address or network space that can be routed.

NAT Inside and Outside Addresses

NAT inside refers to networks owned by an organization that must be translated. When NAT is configured, hosts within this network will have addresses in one space (known as the local address space) that will appear to those outside the network as being in another space (known as the global address space).

Similarly, NAT outside refers to those networks to which the stub network connects. They are not generally under the control of the organization. Hosts in outside networks can be subject to translation and can have local and global addresses.

NAT uses the following definitions:

- Local address—A local IP address that appears on the inside of a network.
- Global address—A global IP address that appears on the outside of a network.
- Inside local address—The IP address that is assigned to a host on the inside network. The address is probably not a legitimate IP address assigned by the Internet Network Information Center (InterNIC) or a service provider.
- Inside global address—A legitimate IP address (assigned by InterNIC or a service provider) that represents one or more inside local IP addresses to the outside world.
- Outside local address—The IP address of an outside host as it appears to the inside network. The address is not necessarily legitimate; it was allocated from the address space that is routable on the inside.
- Outside global address—The IP address that is assigned to a host on the outside network by the owner of the host. The address was allocated from a globally routable address or a network space.

Guidelines and Limitations for Static NAT

Static NAT has the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- If the translated IP is part of the outside interface subnet, then use the **ip proxy-arp** command on the NAT outside interface. If the **add-route** keyword is used, **ip proxy-arp** should be enabled.
- The Cisco Nexus device supports NAT on the following interface types:
    - Routed ports
- NAT is supported on the default Virtual Routing and Forwarding (VRF) table only.
- NAT is supported for IPv4 Unicast only.
- The Cisco Nexus device does not support the following:
    - Software translation. All translations are done in the hardware.
    - NAT routing.
    - Application layer translation. Layer 4 and other embedded IPs are not translated, including FTP, ICMP failures, IPSec, and HTTPs.

  - NAT and VLAN Access Control Lists (VACLs) that are configured on an interface at the same time.
  - PAT translation of fragmented IP packets.
  - NAT translation on software forwarded packets. For example, packets with IP-options are not NAT translated.

- If an IP address is used for Static NAT or PAT translations, it cannot be used for any other purpose. For example, it cannot be assigned to an interface.
- For Static NAT, the outside global IP address should be different from the outside interface IP address.
- When configuring a large number of translations (more than 100), it is faster to configure the translations before configuring the NAT interfaces.
- ECMP NAT is not supported on Cisco Nexus® 3550-T switches.

Configuring Static NAT

Enabling Static NAT

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature nat**	Enables the static NAT feature on the device.
Step 3	switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configuring Static NAT on an Interface

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **ip nat** {**inside** | **outside**}	Specifies the interface as inside or outside. **Note** Only packets that arrive on a marked interface can be translated.

	Command or Action	Purpose
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure an interface with static NAT from the inside:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# ip nat inside
```

Enabling Static NAT for an Inside Source Address

For inside source translation, the traffic flows from inside interface to the outside interface. NAT translates the inside local IP address to the inside global IP address. On the return traffic, the destination inside global IP address gets translated back to the inside local IP address.

Note When the Cisco Nexus device is configured to translate an inside source IP address (Src:ip1) to an outside source IP address (newSrc:ip2), the Cisco Nexus device implicitly adds a translation for an outside destination IP address (Dst: ip2) to an inside destination IP address (newDst: ip1).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip nat inside source static** *local-ip-address global-ip-address* [**group** *group-id*]	Configures static NAT to translate the inside local address to the inside global address or to translate the opposite (the inside global traffic to the inside local traffic). Specifying **group** specifies the group to which this translation belongs on the static twice NAT.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure static NAT for an inside source address:

```
switch# configure terminal
switch(config)# ip nat inside source static 1.1.1.1 5.5.5.5
switch(config)# copy running-config startup-config
```

Enabling Static NAT for an Outside Source Address

For outside source translation, the traffic flows from the outside interface to the inside interface. NAT translates the outside global IP address to the outside local IP address. On the return traffic, the destination outside local IP address gets translated back to outside global IP address.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip nat outside source static** *outsideGlobalIP outsideLocalIP* [**dynamic**] [**add-route**]]	Configures static NAT to translate the outside global address to the outside local address or to translate the opposite (the outside local traffic to the outside global traffic). When an inside translation without ports is configured, an implicit add route is performed. The original add route functionality is an option while configuring an outside translation.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example show how to configure static NAT for an outside source address:

```
switch# configure terminal
switch(config)# ip nat outside source static 2.2.2.2 6.6.6.6
switch(config)# copy running-config startup-config
```

Configuring Static PAT for an Inside Source Address

You can map services to specific inside hosts using Port Address Translation (PAT).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip nat inside source static** {*inside-local-address inside-global-address* | {**tcp**| **udp**} *inside-local-address* {*local-tcp-port*	Maps static NAT to an inside local port to an inside global port.

	Command or Action	Purpose
	| *local-udp-port*} *inside-global-address* {*global-tcp-port* | *global-udp-port*}}	
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to map UDP services to a specific inside source address and UDP port:

```
switch# configure terminal
switch(config)#  ip nat inside source static udp 20.1.9.2 63 35.48.35.48 130
switch(config)# copy running-config startup-config
```

Configuring Static PAT for an Outside Source Address

You can map services to specific outside hosts using Port Address Translation (PAT).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip nat outside source static** {*outside-global-address outside-local-address* | {**tcp** | **udp**} *outside-global-address* {*global-tcp-port* | *global-udp-port*} *outside-local-address* {*global-tcp-port* | *global-udp-port*}} {**add-route**}	Maps static NAT to an outside global port to an outside local port.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to map TCP services to a specific outside source address and TCP port:

```
switch# configure terminal
switch(config)#  ip nat outside source static tcp 20.1.9.2 63 35.48.35.48 130
switch(config)# copy running-config startup-config
```

Enabling and Disabling no-alias Configuration

NAT devices own Inside Global (IG) and Outside Local (OL) addresses and they are responsible for responding to any ARP requests directed to these addresses. When the IG/OL address subnet matches with the local interface subnet, NAT installs an IP alias and an ARP entry, in this case the device uses local-proxy-arp to respond to ARP requests.

The *no-alias* feature responds to ARP requests of all the translated IPs from a given NAT pool address range if the address range is in same subnet of the outside interface.

If no-alias is enabled on an interface with NAT configuration, the outside interface will not respond to any ARP requests in its subnet. When no-alias is disabled, the ARP requests for IPs in same subnet as of outside interface are served.

Note When you downgrade to any older releases that does not support this feature, configurations with *no-alias* option may be deleted.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature nat**	Enables the static NAT feature on the device.
Step 3	switch(config)# **show run nat**	Displays NAT configuration.
Step 4	switch(config)# **show ip nat-alias**	Displays the information whether or not the alias is created. **Note** By default, alias is created. To disable the alias, you must append *no-alias* keyword to the command.
Step 5	switch(config)# **clear ip nat-alias** *ip address/all*	Removes entries from alias list. To remove a specific entry you must provide the IP address that you want to remove. To remove all entries, use the all keyword.

Example

This example shows the interface information:

```
switch# configure terminal
switch(config)# show ip int b
IP Interface Status for VRF "default"(1)
Interface            IP Address      Interface Status
Lo0                  100.1.1.1       protocol-up/link-up/admin-up
Eth1/1               7.7.7.1         protocol-up/link-up/admin-up
Eth1/3               8.8.8.1         protocol-up/link-up/admin-up
```

This example shows the running configuration:

```
switch# configure terminal
switch(config)# show running-config nat
!Command: show running-config nat
!Running configuration last done at: Thu Aug 23 11:57:01 2018
!Time: Thu Aug 23 11:58:13 2018
```

```
version 9.2(2) Bios:version 07.64
feature nat
interface Ethernet1/1
  ip nat inside
interface Ethernet1/3
  ip nat outside
switch(config)#
```

This example shows how to configure alias:

```
switch# configure terminal
switch(config)# ip nat pool p1 7.7.7.2 7.7.7.20 prefix-length 24
switch(config)# ip nat inside source static 1.1.1.2 8.8.8.3
switch(config)# ip nat outside source static 2.2.2.1 7.7.7.3
switch(config)# show ip nat-alias
Alias Information for Context: default
Address          Interface
7.7.7.2           Ethernet1/1
8.8.8.2           Ethernet1/3
switch(config)#
```

This example shows the output of *show ip nat-alias*. By default, alias is enabled.

```
switch# configure terminal
switch(config)# show ip nat-alias
Alias Information for Context: default
Address          Interface
7.7.7.2           Ethernet1/1
8.8.8.2           Ethernet1/3
switch(config)#
```

This example shows how to disable alias:

```
switch# configure terminal
switch(config)# ip nat pool p1 7.7.7.2 7.7.7.20 prefix-length 24 no-alias
switch(config)# ip nat inside source static 1.1.1.2 8.8.8.3 no-alias
switch(config)# ip nat outside source static 2.2.2.1 7.7.7.3 no-alias
switch(config)# show ip nat-alias
Alias Information for Context: default
Address          Interface
7.7.7.2           Ethernet1/1
8.8.8.2           Ethernet1/3
switch(config)#

** None of the entry got appended as alias is disabled for above CLIs.
switch(config)#
```

This example shows how to clear alias. Use *clear ip nat-alias* to remove an entry from alias list. You can remove a single entry by specifying the IP address or remove all the alias entries.

```
switch# configure terminal
switch(config)# clear ip nat-alias address 7.7.7.2
switch(config)# show ip nat-alias
Alias Information for Context: default
Address          Interface
8.8.8.2           Ethernet1/3
switch(config)#
switch(config)# clear ip nat-alias all
switch(config)# show ip nat-alias
switch(config)#
```

Configuration Example for Static NAT and PAT

This example shows the configuration for static NAT:

```
ip nat inside source static 103.1.1.1 11.3.1.1
ip nat inside source static 139.1.1.1 11.39.1.1
ip nat inside source static 141.1.1.1 11.41.1.1
ip nat inside source static 149.1.1.1 95.1.1.1
ip nat inside source static 149.2.1.1 96.1.1.1
ip nat outside source static 95.3.1.1 95.4.1.1
ip nat outside source static 96.3.1.1 96.4.1.1
ip nat outside source static 102.1.2.1 51.1.2.1
ip nat outside source static 104.1.1.1 51.3.1.1
ip nat outside source static 140.1.1.1 51.40.1.1
```

This example shows the configuration for static PAT:

```
ip nat inside source static tcp 10.11.1.1 1 210.11.1.1 101
ip nat inside source static tcp 10.11.1.1 2 210.11.1.1 201
ip nat inside source static tcp 10.11.1.1 3 210.11.1.1 301
ip nat inside source static tcp 10.11.1.1 4 210.11.1.1 401
ip nat inside source static tcp 10.11.1.1 5 210.11.1.1 501
ip nat inside source static tcp 10.11.1.1 6 210.11.1.1 601
ip nat inside source static tcp 10.11.1.1 7 210.11.1.1 701
ip nat inside source static tcp 10.11.1.1 8 210.11.1.1 801
ip nat inside source static tcp 10.11.1.1 9 210.11.1.1 901
ip nat inside source static tcp 10.11.1.1 10 210.11.1.1 1001
ip nat inside source static tcp 10.11.1.1 11 210.11.1.1 1101
ip nat inside source static tcp 10.11.1.1 12 210.11.1.1 1201
```

Verifying the Static NAT Configuration

To display the static NAT configuration, perform this task:

Procedure

	Command or Action	Purpose
Step 1	switch# **show ip nat translations**	Shows the translations for the inside global, inside local, outside local, and outside global IP addresses.

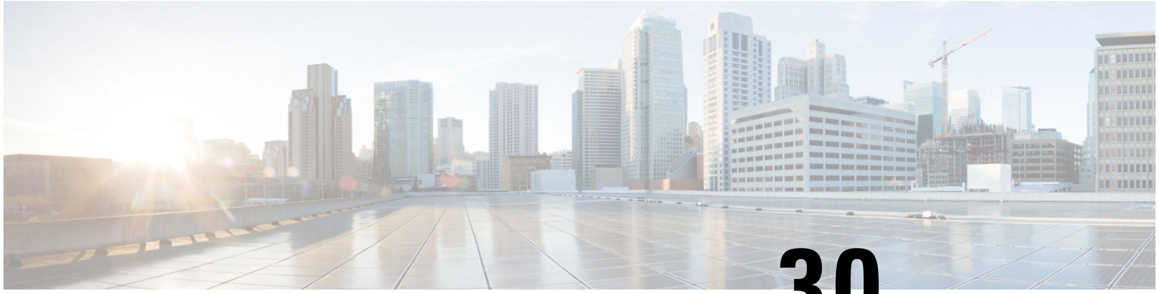
Example

This example shows how to display the static NAT configuration:

```
switch# sh ip nat translations
Pro Inside global      Inside local       Outside local      Outside global

--- ---                ---                51.3.1.1           104.1.1.1
--- ---                ---                95.4.1.1           95.3.1.1
--- ---                ---                96.4.1.1           96.3.1.1
--- ---                ---                51.40.1.1          140.1.1.1
--- ---                ---                51.42.1.1          142.1.2.1
--- ---                ---                51.1.2.1           102.1.2.1
--- 11.1.1.1           101.1.1.1          ---                ---
--- 11.3.1.1           103.1.1.1          ---                ---
--- 11.39.1.1          139.1.1.1          ---                ---
--- 11.41.1.1          141.1.1.1          ---                ---
--- 95.1.1.1           149.1.1.1          ---                ---
--- 96.1.1.1           149.2.1.1          ---                ---
    130.1.1.1:590      30.1.1.100:5000    ---                ---
    130.2.1.1:590      30.2.1.100:5000    ---                ---
    130.3.1.1:590      30.3.1.100:5000    ---                ---
    130.4.1.1:590      30.4.1.100:5000    ---                ---
    130.1.1.1:591      30.1.1.101:5000    ---                ---


switch# sh ip nat translations verbose
Pro Inside global      Inside local       Outside local      Outside global
any ---                ---                22.1.1.3           22.1.1.2
    Flags:0x200009 time-left(secs):-1 id:0 state:0x0 grp_id:10
any 11.1.1.130         11.1.1.3           ---                ---
    Flags:0x1 time-left(secs):-1 id:0 state:0x0 grp_id:0
any 11.1.1.133         11.1.1.33          ---                ---
    Flags:0x1 time-left(secs):-1 id:0 state:0x0 grp_id:10
any 11.1.1.133         11.1.1.33          22.1.1.3           22.1.1.2
    Flags:0x200009 time-left(secs):-1 id:0 state:0x0 grp_id:0
tcp 10.1.1.100:64490   10.1.1.2:0         20.1.1.2:0         20.1.1.2:0
    Flags:0x82 time-left(secs):43192 id:31 state:0x3 grp_id:0 vrf: default
N3550T-1#
```



CHAPTER 30

Configuring Layer 2 Interfaces

This chapter describes how to configure Layer 2 switching ports as access or trunk ports on Cisco NX-OS devices.

Note A Layer 2 port can function as either one of the following:

- A trunk port
- An access port

Note See the System Management Overview, on page 129 for information about configuring a SPAN destination interface.

You can configure Layer 2 switching ports as access or trunk ports. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network. All Layer 2 switching ports maintain media access control (MAC) address tables.

Note See the Layer 2 Switching Configuration Guide, on page 379 for information about VLANs, MAC address tables, private VLANs, and the Spanning Tree Protocol.

Information About Access and Trunk Interfaces

Note The device supports only IEEE 802.1Q-type VLAN trunk encapsulation.

About Access and Trunk Interfaces

A Layer 2 port can be configured as an access or a trunk port as follows:

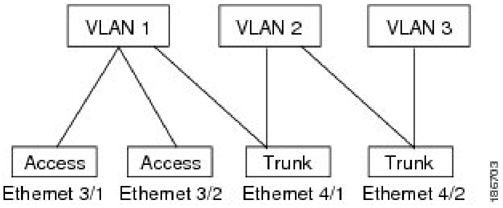
- An access port can have only one VLAN configured on that port; it can carry traffic for only one VLAN.
- A trunk port can have two or more VLANs configured on that port; it can carry traffic for several VLANs simultaneously.

By default, all the ports on the Cisco Nexus® 3550-T switches are Layer 3 ports/Layer 2 ports.

You can make all ports Layer 2 ports using the setup script or by entering the **system default switchport**command. See the *Cisco Nexus® 3550-T Fundamentals Configuration* section for information about using the setup script. To configure the port as a Layer 2 port using the CLI, use the **switchport** command.

The following figure shows how you can use trunk ports in the network. The trunk port carries traffic for two or more VLANs.

Figure 26: Trunk and Access Ports and VLAN Traffic



Note See the *Cisco Nexus® 3550-T Layer 2 Switching Configuration* section for information about VLANs.

In order to correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation, or tagging, method (see the “IEEE 802.1Q Encapsulation” section for more information).

To optimize the performance on access ports, you can configure the port as a host port. Once the port is configured as a host port, it is automatically set as an access port, and channel grouping is disabled. Use the host designation to decrease the time that it takes the designated port to begin to forward packets.

Only an end station can be set as a host port; you will receive an error message if you attempt to configure other ports as hosts.

If an access port receives a packet with an 802.1Q tag in the header other than the access VLAN value, that port drops the packet without learning its MAC source address.

A Layer 2 interface can function as either an access port or a trunk port; it cannot function as both port types simultaneously.

When you change a Layer 2 interface back to a Layer 3 interface, that interface loses all the Layer 2 configuration and resumes the default VLAN configurations.

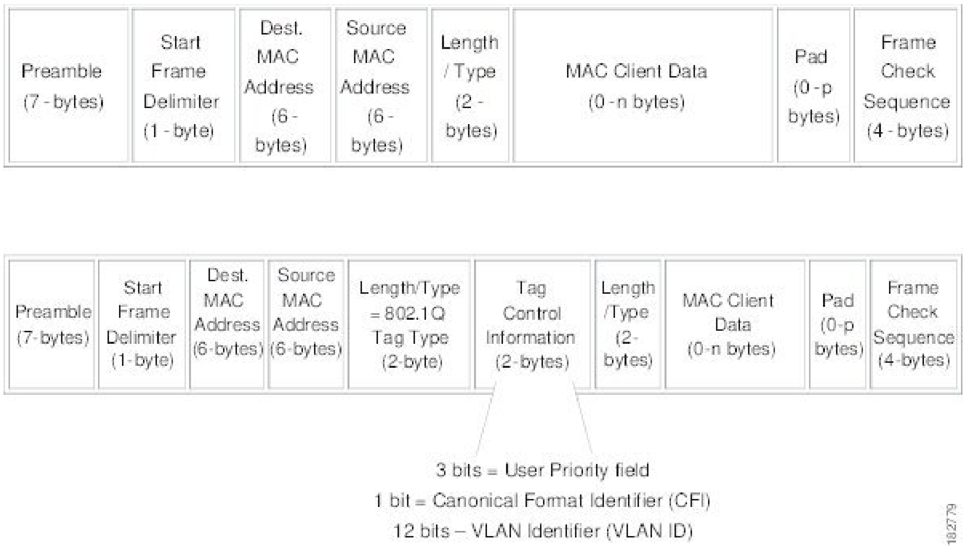
IEEE 802.1Q Encapsulation

Note For information about VLANs, see the *Cisco Nexus® 3550-T Layer 2 Switching Configuration* section.

A trunk is a point-to-point link between the switch and another networking device. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network.

To correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation, or tagging, method that uses a tag that is inserted into the frame header. This tag carries information about the specific VLAN to which the frame and packet belong. This method allows packets that are encapsulated for several different VLANs to traverse the same port and maintain traffic separation between the VLANs. Also, the encapsulated VLAN tag allows the trunk to move traffic end-to-end through the network on the same VLAN.

Figure 27: Header Without and With 802.1Q Tag



Preamble (7 - bytes)	Start Frame Delimiter (1 - byte)	Dest. MAC Address (6 - bytes)	Source MAC Address (6 - bytes)	Length / Type (2 - bytes)	MAC Client Data (0 - n bytes)	Pad (0 - p bytes)	Frame Check Sequence (4 - bytes)

Preamble (7-bytes)	Start Frame Delimiter (1-byte)	Dest. MAC Address (6-bytes)	Source MAC Address (6-bytes)	Length/Type = 802.1Q Tag Type (2-byte)	Tag Control Information (2-bytes)	Length /Type (2-bytes)	MAC Client Data (0-n bytes)	Pad (0-p bytes)	Frame Check Sequence (4-bytes)

3 bits = User Priority field
1 bit = Canonical Format Identifier (CFI)
12 bits – VLAN Identifier (VLAN ID)
182779

Access VLANs

When you configure a port in access mode, you can specify which VLAN will carry the traffic for that interface. If you do not configure the VLAN for a port in access mode, or an access port, the interface carries traffic for the default VLAN (VLAN1).

You can change the access port membership in a VLAN by specifying the new VLAN. You must create the VLAN before you can assign it as an access VLAN for an access port. If you change the access VLAN on an access port to a VLAN that is not yet created, the system shuts that access port down.

If an access port receives a packet with an 802.1Q tag in the header other than the access VLAN value, that port drops the packet without learning its MAC source address.

Native VLAN IDs for Trunk Ports

A trunk port can carry nontagged packets simultaneously with the 802.1Q tagged packets. When you assign a default port VLAN ID to the trunk port, all untagged traffic travels on the default port VLAN ID for the trunk port, and all untagged traffic is assumed to belong to this VLAN. This VLAN is referred to as the native VLAN ID for a trunk port. That is, the native VLAN ID is the VLAN that carries untagged traffic on trunk ports.

Note Native VLAN ID numbers must match on both ends of the trunk.

The trunk port sends an egressing packet with a VLAN that is equal to the default port VLAN ID as untagged; all the other egressing packets are tagged by the trunk port. If you do not configure a native VLAN ID, the trunk port uses the default VLAN.

Tagging Native VLAN Traffic

The Cisco software supports the IEEE 802.1Q standard on trunk ports. In order to pass untagged traffic through the trunk ports, you must create a VLAN that does not tag any packets (or you can use the default VLAN). Untagged packets can pass through trunk ports and access ports.

However, all packets that enter the device with an 802.1Q tag that matches the value of the native VLAN on the trunk are stripped of any tagging and egress the trunk port as untagged packets. This situation can cause problems because you may want to retain the tagging on packets on the native VLAN for the trunk port.

Allowed VLANs

By default, a trunk port sends traffic to and receives traffic from all VLANs. All VLAN IDs are allowed on each trunk. However, you can remove VLANs from this inclusive list to prevent traffic from the specified VLANs from passing over the trunk. Later, you can add any specific VLANs that you may want the trunk to carry traffic for back to the list.

To partition the Spanning Tree Protocol (STP) topology for the default VLAN, you can remove VLAN1 from the list of allowed VLANs. Otherwise, VLAN1, which is enabled on all ports by default, will have a very big STP topology, which can result in problems during STP convergence. When you remove VLAN1, all data traffic for VLAN1 on this port is blocked, but the control traffic continues to move on the port.

Note See the *Cisco Nexus® 3550-T Layer 2 Switching Configuration* section for more information about STP.

Default Interfaces

You can use the default interface feature to clear the configured parameters for both physical and logical interfaces such as the Ethernet, loopback, VLAN network, and the port-channel interface.

Note All 48 ports can be selected for the default interface.

Switch Virtual Interface and Autostate Behavior

In Cisco NX-OS, a switch virtual interface (SVI) represents a logical interface between the bridging function and the routing function of a VLAN in the device.

The operational state of this interface is governed by the state of the various ports in its corresponding VLAN. An SVI interface on a VLAN comes up when at least one port in that VLAN is in the Spanning Tree Protocol (STP) forwarding state. Similarly, this interface goes down when the last STP forwarding port goes down or goes to another STP state.

High Availability

The software supports high availability for Layer 2 ports.

Note See the *Cisco Nexus® 3550-T High Availability and Redundancy Guide* for complete information about high availability features.

Prerequisites for Layer 2 Interfaces

Layer 2 interfaces have the following prerequisites:

- You are logged onto the device.
- By default, Cisco NX-OS configures Layer 3 parameters. If you want to configure Layer 2 parameters, you need to switch the port mode to Layer 2. You can change the port mode by using the **switchport** command.
- You must configure the port as a Layer 2 port before you can use the **switchport mode**command. By default, all ports on the device are Layer 3 ports. By default, all ports on the Cisco Nexus® 3550-T devices are Layer 2 ports.

Guidelines and Limitations for Layer 2 Interfaces

VLAN trunking has the following configuration guidelines and limitations:

- A port can be either a Layer 2 or a Layer 3 interface; it cannot be both simultaneously.
- When you change a Layer 3 port to a Layer 2 port or a Layer 2 port to a Layer 3 port, all layer-dependent configuration is lost. When you change an access or trunk port to a Layer 3 port, all information about the access VLAN, native VLAN, allowed VLANs, and so forth, is lost.
- Do not connect devices with access links because access links may partition a VLAN.
- When connecting Cisco devices through an 802.1Q trunk, make sure that the native VLAN for an 802.1Q trunk is the same on both ends of the trunk link. If the native VLAN on one end of the trunk is different from the native VLAN on the other end, spanning tree loops might result.

- Disabling spanning tree on the native VLAN of an 802.1Q trunk without disabling spanning tree on every VLAN in the network can cause spanning tree loops. You must leave spanning tree enabled on the native VLAN of an 802.1Q trunk. If you cannot leave spanning tree enabled, you must disable spanning tree on every VLAN in the network. Make sure that your network has no physical loops before you disable spanning tree.
- When you connect two Cisco devices through 802.1Q trunks, the devices exchange spanning tree bridge protocol data units (BPDUs) on each VLAN allowed on the trunks. The BPDUs on the native VLAN of the trunk are sent untagged to the reserved IEEE 802.1D spanning tree multicast MAC address (01-80-C2-00-00-00). The BPDUs on all other VLANs on the trunk are sent tagged to the reserved Cisco Shared Spanning Tree (SSTP) multicast MAC address (01-00-0c-cc-cc-cd).
- Non-Cisco 802.1Q devices maintain only a single instance of spanning tree (the Mono Spanning Tree) that defines the spanning tree topology for all VLANs. When you connect a Cisco switch to a non-Cisco switch through an 802.1Q trunk, the Mono Spanning Tree of the non-Cisco switch and the native VLAN spanning tree of the Cisco switch combine to form a single spanning tree topology known as the Common Spanning Tree (CST).
- Because Cisco devices transmit BPDUs to the SSTP multicast MAC address on VLANs other than the native VLAN of the trunk, non-Cisco devices do not recognize these frames as BPDUs and flood them on all ports in the corresponding VLAN. Other Cisco devices connected to the non-Cisco 802.1Q cloud receive these flooded BPDUs. This BPDU reception allows Cisco switches to maintain a per-VLAN spanning tree topology across a cloud of non-Cisco 802.1Q devices. The non-Cisco 802.1Q cloud that separates the Cisco devices is treated as a single broadcast segment between all devices connected to the non-Cisco 802.1Q cloud through 802.1Q trunks.
- Make certain that the native VLAN is the same on all of the 802.1Q trunks that connect the Cisco devices to the non-Cisco 802.1Q cloud.
- If you are connecting multiple Cisco devices to a non-Cisco 802.1Q cloud, all of the connections must be through 802.1Q trunks. You cannot connect Cisco devices to a non-Cisco 802.1Q cloud through access ports because doing so places the access port on the Cisco device into the spanning tree “port inconsistent” state and no traffic will pass through the port.
- You can group trunk ports into port-channel groups, but all trunks in the group must have the same configuration. When a group is first created, all ports follow the parameters set for the first port to be added to the group. If you change the configuration of one of these parameters, the device propagates that setting to all ports in the group, such as the allowed VLANs and the trunk status. For example, if one port in a port group ceases to be a trunk, all ports cease to be trunks.
- When MAC addresses are cleared on a VLAN with the clear mac address-table dynamic command, the dynamic ARP (Address Resolution Protocol) entries on that VLAN are refreshed.
- If a static ARP entry exists on the VLAN and no MAC address to port mapping is present, the supervisor may generate an ARP request to learn the MAC address. Upon learning the MAC address, the adjacency entry points to the correct physical port.
- Cisco NX-OS does not support transparent bridging between two VLANs when one of the SVIs is on the Cisco Nexus 9000 using the BIA MAC (burned-in MAC address). This occurs when the BIA MAC is shared between SVIs/VLANs. A MAC, different from the BIA MAC, can be configured under the SVI for transparent bridging to work properly.
- You may get an error message when you attempt to configure the interface mode to trunk and trunk VLANs simultaneously. On Cisco NX-OS interfaces, the default value of interface mode is access. To

implement any trunk related configurations, you must first change the interface mode to trunk and then configure the trunk VLAN ranges.

- ***Cisco Nexus 3550-T - 10.1(2t) release*** switch does cut-through forwarding; hence there is no MTU-check implemented.

  Hardware buffering is not designed for jumbo packets and packets beyond regular mtu size 1516 is not supported.

Default Settings for Layer 2 Interfaces

The following table lists the default settings for device access and trunk port mode parameters.

Configuring Access and Trunk Interfaces

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Configuring a VLAN Interface as a Layer 2 Access Port

You can configure a Layer 2 port as an access port. An access port transmits packets on only one, untagged VLAN. You specify which VLAN traffic that the interface carries, which becomes the access VLAN. If you do not specify a VLAN for an access port, that interface carries traffic only on the default VLAN. The default VLAN is VLAN1.

The VLAN must exist before you can specify that VLAN as an access VLAN. The system shuts down an access port that is assigned to an access VLAN that does not exist.

Before you begin

Ensure that you are configuring a Layer 2 interface.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** {{*type slot/port*} | {**port-channel** *number*}}
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Specifies an interface to configure, and enters interface configuration mode.

	Command or Action	Purpose
Step 3	**switchport mode** [**access** | **trunk**] **Example:** `switch(config-if)# switchport mode access`	Sets the interface as a nontrunking nontagged, single-VLAN Layer 2 interface. An access port can carry traffic in one VLAN only. By default, an access port carries traffic for VLAN1; to set the access port to carry traffic for a different VLAN, use the **switchport access vlan** command.
Step 4	**switchport access vlan** *vlan-id* **Example:** `switch(config-if)# switchport access vlan 5`	Specifies the VLAN for which this access port will carry traffic. If you do not enter this command, the access port carries traffic on VLAN1 only; use this command to change the VLAN for which the access port carries traffic.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface configuration mode.

Step 6	**show interface** **Example:** `switch# show interface`	(Optional) Displays the interface status and information.

Step 7	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e1/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set Ethernet 1/1 as a Layer 2 access port that carries traffic for VLAN 5 only:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 5
switch(config-if)#
```

Configuring Access Host Ports

Note You should apply the switchport host command only to interfaces that are connected to an end station.

You can optimize the performance of access ports that are connected to end stations by simultaneously setting that port as an access port. An access host port handles the STP like an edge port and immediately moves to the forwarding state without passing through the blocking and learning states. Configuring an interface as an access host port also disables port channeling on that interface.

Note See the *Configuring Port Channels* and the *Cisco Nexus® 3550-T Layer 2 Switching Configuration* sections for information about port-channel interfaces

Before you begin

Ensure that you are configuring the correct interface to an interface that is an end station.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**switchport host** **Example:** switch(config-if)# `switchport host`	Sets the interface to be an access host port, which immediately moves to the spanning tree forwarding state and disables port channeling on this interface. **Note** Apply this command only to end stations.

Step 4	**exit**
	Example:
	`switch(config-if-range)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 5	**show interface** **Example:** `switch# show interface`	(Optional) Displays the interface status and information.

	Command or Action	Purpose

Step 6	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e1/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set Ethernet 1/1 as a Layer 2 access port with PortFast enabled and port channel disabled:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport host
switch(config-if)#
```

Configuring Trunk Ports

You can configure a Layer 2 port as a trunk port. A trunk port transmits untagged packets for one VLAN plus encapsulated, tagged, packets for multiple VLANs. (See the *IEEE 802.1Q Encapsulation* section for information about encapsulation.)

Note The device supports 802.1Q encapsulation only.

Before you begin

Before you configure a trunk port, ensure that you are configuring a Layer 2 interface.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** {*type slot/port* | **port-channel** *number*} **Example:**	Specifies an interface to configure, and enters interface configuration mode.

	Command or Action	Purpose

	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`	

Step 3	**switchport mode** [**access** | **trunk**] **Example:** `switch(config-if)# switchport mode trunk`	Sets the interface as a Layer 2 trunk port. A trunk port can carry traffic in one or more VLANs on the same physical link (VLANs are based on the trunk-allowed VLANs list). By default, a trunk interface can carry traffic for all VLANs. To specify that only certain VLANs are allowed on the specified trunk, use the **switchport trunk allowed vlan** command.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 5	**show interface** **Example:** `switch# show interface`	(Optional) Displays the interface status and information.

Step 6	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e1/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set Ethernet 1/1 as a Layer 2 trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport mode trunk
switch(config-if)#
```

Configuring the Native VLAN for 802.1Q Trunking Ports

You can configure the native VLAN for 802.1Q trunk ports. If you do not configure this parameter, the trunk port uses the default VLAN as the native VLAN ID.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** {{*type slot/port*} | {**port-channel** *number*}}
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**switchport trunk native vlan** *vlan-id* **Example:** `switch(config-if)# switchport trunk native vlan 5`	Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 4094, except those VLANs reserved for internal use. The default value is VLAN1.

Step 4	**exit**
	Example:
	`switch(config-if-range)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 5	**show vlan** **Example:** `switch# show vlan`	(Optional) Displays the status and information of VLANs.

Step 6	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e1/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set the native VLAN for the Ethernet 1/1, Layer 2 trunk port to VLAN 5:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport trunk native vlan 5
switch(config-if)#
```

Configuring the Allowed VLANs for Trunking Ports

You can specify the IDs for the VLANs that are allowed on the specific trunk port.

> **Note** The **switchport trunk allowed vlan** *vlan-list* command replaces the current VLAN list on the specified port with the new list. You are prompted for confirmation before the new list is applied.
>
> If you are doing a copy and paste of a large configuration, you might see some failures because the CLI is waiting for a confirmation before accepting other commands. To avoid this problem, you can disable prompting by using the **terminal dont-ask** command before you paste the configuration.

Before you begin

Before you configure the allowed VLANs for the specified trunk ports, ensure that you are configuring the correct interfaces and that the interfaces are trunks.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** {**ethernet** *slot/port* | **port-channel** *number*}
	Example:
	`switch(config)# interface ethernet 1/1`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**switchport trunk allowed vlan** {*vlan-list* **add** *vlan-list* | **all** | **except** *vlan-list* | **none** | **remove** *vlan-list*} **Example:** `switch(config-if)# switchport trunk allowed vlan add 15-20#`	Sets the allowed VLANs for the trunk interface. The default is to allow all VLANs on the trunk interface: 1 to 3967 and 4048 to 4094.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode.

Step 5	**show vlan** **Example:** `switch# show vlan`	(Optional) Displays the status and information for VLANs.
Step 6	**no shutdown** **Example:**	(Optional) Clears the errors on the interfaces and VLANs where policies correspond with

	Command or Action	Purpose

	`switch# configure terminal`
	`switch(config)# int e1/1`
	`switch(config-if)# no shutdown`
		hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to add VLANs 15 to 20 to the list of allowed VLANs on the Ethernet 1/1, Layer 2 trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport trunk allowed vlan 15-20
switch(config-if)#
```

Configuring a Default Interface

The default interface feature allows you to clear the existing configuration of multiple interfaces such as Ethernet, loopback, VLAN network, port-channel, and tunnel interfaces. All user configuration under a specified interface will be deleted. You can optionally create a checkpoint before clearing the interface configuration so that you can later restore the deleted configuration.

Note The default interface feature is not supported for management interfaces because the device could go to an unreachable state.

Procedure

	Command or Action	Purpose

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

Step 2	**default interface** *int-if* [**checkpoint** *name*] **Example:** `switch(config)# default interface ethernet 1/1 checkpoint test8`	Deletes the configuration of the interface and restores the default configuration. Use the **?** keyword to display the supported interfaces. Use the **checkpoint** keyword to store a copy of the running configuration of the interface before clearing the configuration.

	Command or Action	Purpose

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch(config)#`
		Exits global configuration mode.

Step 4	**show interface** **Example:** `switch# show interface`	(Optional) Displays the interface status and information.

Step 5	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e1/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Example

This example shows how to delete the configuration of an Ethernet interface while saving a checkpoint of the running configuration for rollback purposes:

```
switch# configure terminal
switch(config)# default interface ethernet 1/1 checkpoint test8
.......Done
switch(config)#
```

Changing the System Default Port Mode to Layer 2

You can set the system default port mode to Layer 2 access ports.

Procedure

	Command or Action	Purpose

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

Step 2	**system default switchport** [**shutdown**] **Example:** switch(config-if)# `system default switchport`	Sets the default port mode for all interfaces on the system to Layer 2 access port mode and enters interface configuration mode. By default, all the interfaces are Layer 3.

	Command or Action	Purpose
		Note When the **system default switchport shutdown** command is issued: • Any Layer 2 port that is not specifically configured with **no shutdown** are shutdown. To avoid the shutdown, configure the Layer 2 port with **no shut**

Step 3	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface configuration mode.

Step 4	**show interface brief** **Example:** `switch# show interface brief`	(Optional) Displays the status and information for interfaces.

Step 5	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e1/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set the system ports to be Layer 2 access ports by default:

```
switch# configure terminal
switch(config-if)# system default switchport
switch(config-if)#
```

Verifying the Interface Configuration

To display access and trunk interface configuration information, perform one of the following tasks.

Command	Purpose
show interface ethernet *slot/port* [**brief** | | **counters** | **debounce** | **description** | **flowcontrol** | **mac-address** | **status** | **transceiver**]	Displays the interface configuration.
show interface brief	Displays interface configuration information, including the mode.
show interface switchport	Displays information, including access and trunk interface, information for all Layer 2 interfaces.
show interface trunk [**module** *module-number* | **vlan** *vlan-id*]	Displays trunk configuration information.
show interface capabilities	Displays information about the capabilities of the interfaces.
show running-config [**all**]	Displays information about the current configuration. The **all** command displays the default and current configurations.
show running-config interface ethernet *slot/port*	Displays configuration information about the specified interface.
show running-config interface port-channel *slot/port*	Displays configuration information about the specified port-channel interface.
show running-config interface vlan *vlan-id*	Displays configuration information about the specified VLAN interface.

Monitoring the Layer 2 Interfaces

Use the following commands to display Layer 2 interfaces:

Command	Purpose
clear counters interface [**interface**]	Clears the counters.
show interface counters [**module** *module*]	Displays input and output octets unicast packets, multicast packets, and broadcast packets.
show interface counters detailed [**all**]	Displays input packets, bytes, and multicast as well as output packets and bytes.
show interface counters errors [**module** *module*]	Displays information on the number of error packets.

Configuration Examples for Access and Trunk Ports

This example shows how to configure a Layer 2 access interface and assign the access VLAN mode for that interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/30
switch(config-if)# switchport
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 5
switch(config-if)#
```

This example shows how to configure a Layer 2 trunk interface, assign the native VLAN and the allowed VLANs, and configure the device to tag the native VLAN traffic on the trunk interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/35
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk native vlan 10
switch(config-if)# switchport trunk allowed vlan 5, 10
switch(config-if)# exit
switch(config)#
```

Related Documents

Related Documents	**Document Title**
Configuring Layer 3 interfaces	*Configuring Layer 2 Interfaces* section
Port Channels	*Configuring Port Channels* section
VLANs, and STP	*Cisco Nexus® 3550-T Layer 2 Switching Configuration* chapter
System management	*Cisco Nexus® 3550-T System Management Configuration* chapter
High availability	*Cisco Nexus® Series High Availability and Redundancy Guide*
Licensing	*Cisco NX-OS Licensing Guide*
Release Notes	*Cisco Nexus® Series NX-OS Release Notes*

CHAPTER 31

Configuring Port Channels

About Port Channels

A port channel is an aggregation of multiple physical interfaces that creates a logical interface. You can bundle up to 4 individual active links into a port channel to provide increased bandwidth and redundancy. Port channeling also load balances traffic across these physical interfaces. The port channel stays operational as long as at least one physical interface within the port channel is operational.

You can create a Layer 2 port channel by bundling compatible Layer 2 interfaces, or you can create Layer 3 port channels by bundling compatible Layer 3 interfaces. You cannot combine Layer 2 and Layer 3 interfaces in the same port channel.

You can also change the port channel from Layer 3 to Layer 2. See the *Configuring Layer 2 Interfaces* chapter for information about creating Layer 2 interfaces.

A Layer 2 port channel interface and it's member ports can have different STP parameters. Changing the STP parameters of the port channel does not impact the STP parameters of the member ports because a port channel interface takes precedence if the member ports are bundled.

Note Members can be bundled into a port channel only if they belong to same **Quad**.

Note After a Layer 2 port becomes part of a port channel, all switchport configurations must be done on the port channel; you can no longer apply switchport configurations to individual port-channel members. You cannot apply Layer 3 configurations to an individual port-channel member either; you must apply the configuration to the entire port channel.

You can use static port channels, with no associated aggregation protocol, for a simplified configuration.

For more flexibility, you can use the Link Aggregation Control Protocol (LACP), which is defined in IEEE 802.3ad. When you use LACP, the link passes protocol packets. You cannot configure LACP on shared interfaces.

See the *LACP Overview* section for information about LACP.

Port Channels

A port channel bundles physical links into a channel group to create a single logical link that provides the aggregate bandwidth of up to 4 physical links. If a member port within a port channel fails, the traffic previously carried over the failed link switches to the remaining member ports within the port channel.

However, you can enable the LACP to use port channels more flexibly. Configuring port channels with LACP and static port channels require a slightly different procedure (see the *Configuring Port Channels* section).

Note The device does not support Port Aggregation Protocol (PAgP) for port channels.

Each port can be in only one port channel. All the ports in a port channel must be compatible; they must use the same speed and duplex mode (see the *Compatibility Requirements* section). When you run static port channels with no aggregation protocol, the physical links are all in the on channel mode; you cannot change this mode without enabling LACP (see the *Port-Channel Modes* section).

You can create port channels directly by creating the port-channel interface, or you can create a channel group that acts to aggregate individual ports into a bundle. When you associate an interface with a channel group, the software creates a matching port channel automatically if the port channel does not already exist. In this instance, the port channel assumes the Layer 2 or Layer 3 configuration of the first interface. You can also create the port channel first. In this instance, the Cisco NX-OS software creates an empty channel group with the same channel number as the port channel and takes the default Layer 2 or Layer 3 configuration, as well as the compatibility configuration (see the *Compatibility Requirements* section).

Note The port channel is operationally up when at least one of the member ports is up and that port’s status is channeling. The port channel is operationally down when all member ports are operationally down.

Port-Channel Interfaces

The following shows port-channel interfaces.

You can classify port-channel interfaces as Layer 2 or Layer 3 interfaces. In addition, you can configure Layer 2 port channels in either access or trunk mode. Layer 3 port-channel interfaces have routed ports as channel members.

You can configure a Layer 3 port channel with a static MAC address. If you do not configure this value, the Layer 3 port channel uses the router MAC of the first channel member to come up. See the *Cisco Nexus® 3550-T Layer 2 Switching Configuration* section for information about configuring static MAC addresses on Layer 3 port channels.

See the *Cisco Nexus® 3550-T Layer 2 Interfaces Configuration* chapter for information about configuring Layer 2 ports in access or trunk mode and the *Configuring Layer 3 Interfaces* chapter for information about configuring Layer 3 interfaces and subinterfaces.

Basic Settings

You can configure the following basic settings for the port-channel interface:

- Bandwidth—Use this setting for informational purposes only; this setting is to be used by higher-level protocols.
- Delay—Use this setting for informational purposes only; this setting is to be used by higher-level protocols.
- Description
- IP addresses
- Shutdown

Compatibility Requirements

When you add an interface to a channel group, the software checks certain interface attributes to ensure that the interface is compatible with the channel group. For example, you cannot add a Layer 3 interface to a Layer 2 channel group. The Cisco NX-OS software also checks a number of operational attributes for an interface before allowing that interface to participate in the port-channel aggregation.

The compatibility check includes the following operational attributes:

- Network layer
- Port mode
- Access VLAN
- Trunk native VLAN
- Tagged or untagged
- Allowed VLAN list
- Flow-control capability
- Flow-control configuration
- Media type, either copper or fiber

Use the **show port-channel compatibility-parameters** command to see the full list of compatibility checks that the Cisco NX-OS uses.

You can only add interfaces configured with the channel mode set to on to static port channels, and you can only add interfaces configured with the channel mode as active or passive to port channels that are running LACP. You can configure these attributes on an individual member port. If you configure a member port with an incompatible attribute, the software suspends that port in the port channel.

Alternatively, you can force ports with incompatible parameters to join the port channel if the following parameters are the same:

- Flow-control capability
- Flow-control configuration

When the interface joins a port channel, some of its individual parameters are removed and replaced with the values on the port channel as follows:

- Bandwidth
- Delay
- IP address
- MAC address
- Spanning Tree Protocol

Many interface parameters remain unaffected when the interface joins or leaves a port channel as follows:

- Beacon
- Description
- CDP
- LACP port priority
- Debounce
- Shutdown
- SNMP trap

Note When you delete the port channel, the software sets all member interfaces as if they were removed from the port channel.

See the “LACP Marker Responders” section for information about port-channel modes.

Load Balancing Using Port Channels

The Cisco NX-OS software load balances traffic across all operational interfaces in a port channel by hashing the addresses in the frame to a numerical value that selects one of the links in the channel. Port channels provide load balancing by default. Port-channel load balancing uses MAC addresses, IP addresses, or Layer

4 port numbers to select the link. Port-channel load balancing uses either source or destination addresses or ports, or both source and destination addresses or ports.

You can configure the load- balancing mode to apply to all port channels that are configured on the entire device. You can configure one load-balancing mode for the entire device. You cannot configure the load-balancing method per port channel.

You can configure the type of load-balancing algorithm used. You can choose the load-balancing algorithm that determines which member port to select for egress traffic by looking at the fields in the frame.

The default load-balancing mode for Layer 3 interfaces is the source and destination IP L4 ports, and the default load-balancing mode for non-IP traffic is the source and destination MAC address. Use the **port-channel load-balance** command to set the load-balancing method among the interfaces in the channel-group bundle. The default method for Layer 2 packets is src-dst-mac. The default method for Layer 3 packets is src-dst ip-l4port.

You can configure the device to use one of the following methods to load balance across the port channel:

- Destination MAC address
- Source MAC address
- Source and destination MAC address
- Destination IP address
- Source IP address
- Source and destination IP address

Non-IP and Layer 3 port channels both follow the configured load-balancing method, using the source, destination, or source and destination parameters. For example, when you configure load balancing to use the source IP address, all non-IP traffic uses the source MAC address to load balance the traffic while the Layer 3 traffic load balances the traffic using the source IP address. Similarly, when you configure the destination MAC address as the load-balancing method, all Layer 3 traffic uses the destination IP address while the non-IP traffic load balances using the destination MAC address.

The unicast and multicast traffic is load-balanced across port-channel links based on configured load-balancing algorithm displayed in **show port-channel load-balancing** command output.

The multicast traffic uses the following methods for load balancing with port channels:

- Multicast traffic without Layer 4 information—Source IP address, destination IP address
- Non-IP multicast traffic—Source MAC address, destination MAC address

Note Devices that run Cisco IOS can optimize the behavior of the member ports ASICs if a failure of a single member occurred by running the port-channel hash-distribution command. The Cisco Nexus 3550-T device performs this optimization by default and does not require or support this command. Cisco NX-OS does support the customization of the load-balancing criteria on port channels through the port-channel load-balance command for the entire device.

LACP

LACP allows you to configure up to 4 interfaces into a port channel.

LACP Overview

The Link Aggregation Control Protocol (LACP) for Ethernet is defined in IEEE 802.1AX and IEEE 802.3ad. This protocol controls how physical ports are bundled together to form one logical channel.

Note You must enable LACP before you can use LACP. By default, LACP is disabled. See the *Enabling LACP* section for information about enabling LACP.

The system automatically takes a checkpoint before disabling the feature, and you can roll back to this checkpoint. See the *Cisco Nexus® 3550-T System Management Configuration* section for information about rollbacks and checkpoints.

Individual links can be combined into LACP port channels and channel groups as well as function as individual links.

With LACP, you can bundle up to 4 interfaces in a channel group.

Note When you delete the port channel, the software automatically deletes the associated channel group. All member interfaces revert to their original configuration.

You cannot disable LACP while any LACP configurations are present.

Port-Channel Modes

Individual interfaces in port channels are configured with channel modes. When you run static port channels with no aggregation protocol, the channel mode is always set to **on**. After you enable LACP globally on the device, you enable LACP for each channel by setting the channel mode for each interface to either **active** or **passive**. You can configure channel mode for individual links in the LACP channel group when you are adding the links to the channel group

Note You must enable LACP globally before you can configure an interface in either the **active** or **passive** channel mode.

The following table describes the channel modes.

Table 27: Channel Modes for Individual Links in a Port Channel

Channel Mode	Description
passive	The LACP is enabled on this port channel and the ports are in a passive negotiating state. Ports responds to LACP packets that it receives but does not initiate LACP negotiation.
active	The LACP is enabled on this port channel and the ports are in an active negotiating state. Ports initiate negotiations with other ports by sending LACP packets.
on	The LACP is disabled on this port channel and the ports are in a non-negotiating state. The **on** state of the port channel represents the static mode. The port will not verify or negotiate port channel memberships. If you attempt to change the channel mode to active or passive before enabling LACP, the device displays an error message. When an LACP attempts to negotiate with an interface in the **on** state, it does not receive any LACP packets and becomes an individual link with that interface, it does not join the LACP channel group. The **on** state is the default port-channel mode

Both the passive and active modes allow LACP to negotiate between ports to determine if they can form a port channel based on criteria such as the port speed and the trunking state.The passive mode is useful when you do not know whether the remote system, or partner, supports LACP.

Two devices can form an LACP port channel when their ports are in different LACP modes if the modes are compatible as in the following example:

Table 28: Channel Modes Compatibility

Device 1 > Port-1	Device 2 > Port-2	Result
Active	Active	Can form a port channel.
Active	Passive	Can form a port channel.
Passive	Passive	Cannot form a port channel because no ports can initiate negotiation.
On	Active	Cannot form a port channel because LACP is enabled only on one side.
On	Passive	Cannot form a port channel because LACP is not enabled.

LACP ID Parameters

This section describes the LACP parameters.

LACP System Priority

Each system that runs LACP has an LACP system priority value. You can accept the default value of 32768 for this parameter, or you can configure a value between 1 and 65535. LACP uses the system priority with the MAC address to form the system ID and also uses the system priority during negotiation with other devices. A higher system priority value means a lower priority.

Note The LACP system ID is the combination of the LACP system priority value and the MAC address.

LACP Port Priority

Each port that is configured to use LACP has an LACP port priority. You can accept the default value of 32768 for the LACP port priority, or you can configure a value between 1 and 65535. LACP uses the port priority with the port number to form the port identifier.

LACP uses the port priority to decide which ports should be put in standby mode when there is a limitation that prevents all compatible ports from aggregating and which ports should be put into active mode. A higher port priority value means a lower priority for LACP. You can configure the port priority so that specified ports have a lower priority for LACP and are most likely to be chosen as active links, rather than hot-standby links.

LACP Administrative Key

LACP automatically configures an administrative key value equal to the channel-group number on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port’s ability to aggregate with other ports is determined by these factors:

- Port physical characteristics, such as the data rate and the duplex capability
- Configuration restrictions that you establish

LACP Marker Responders

You can dynamically redistribute the data traffic by using port channels. This redistribution might result from a removed or added link or a change in the load-balancing scheme. Traffic redistribution that occurs in the middle of a traffic flow can cause misordered frames.

LACP uses the Marker Protocol to ensure that frames are not duplicated or reordered due to this redistribution. The Marker Protocol detects when all the frames of a given traffic flow are successfully received at the remote end. LACP sends Marker PDUs on each of the port-channel links. The remote system responds to the Marker PDU once it receives all the frames received on this link prior to the Marker PDU. The remote system then sends a Marker Responder. Once the Marker Responders are received by the local system on all member links of the port channel, the local system can redistribute the frames in the traffic flow with no chance of misordering. The software supports only Marker Responders.

LACP-Enabled and Static Port Channels Differences

The following table summarizes the major differences between port channels with LACP enabled and static port channels.

Table 29: Port Channels with LACP Enabled and Static Port Channels

Configurations	Port Channels with LACP Enabled	Static Port Channels
Protocol applied	Enable globally	Not applicable
Channel mode of links	Can be either: • Active • Passive	Can only be On
Maximum number of links in channel	4	4

LACP Compatibility Enhancements

When a Cisco Nexus 3550-T device is connected to a non-Nexus peer, its graceful failover defaults may delay the time that is taken to bring down a disabled port or cause traffic from the peer to be lost. To address these conditions, the **lacp graceful-convergence** command was added.

By default, LACP sets a port to suspended state if it does not receive an LACP PDU from the peer. **lacp suspend-individual** is a default configuration on Cisco Nexus® 3550-T switches. This command puts the port in suspended state if it does not receive any LACP PDUs. In some cases, although this feature helps in preventing loops created due to misconfigurations, it can cause servers fail to boot up because they require LACP to logically bring up the port. You can put a port into an individual state by using the **no lacp suspend-individual**. Port in individual sate takes attributes of the individual port based on the port configuration.

LACP port-channels exchange LACP PDUs for quick bundling of links when connecting a server and a switch. However, the links go into suspended state when the PDUs are not received.

The **delayed LACP** feature enables one port-channel member, the delayed-LACP port, to come up first as a member of a regular port-channel before LACP PDUs are received. After it is connected in LACP mode, other members, the auxiliary LACP ports, are brought up. This avoids having the links becoming suspended when PDUs are not received.

Which port in the port-channel comes up first depends on the port-priority value of the ports. A member link in a port channel with lowest priority value, will come come up first as a LACP delayed port. Regardless of the operational status of the links, the configured priority of a LACP port is used to select the delayed-lacp port

This feature supports Layer 2 port channels and trunk mode spanning tree and has the following limitations:

- Using **no lacp suspend-individual** and **lacp mode delay** on a same port channel is not recommended because it can put non-lacp delayed ports in individual state. As a best practice, you must avoid combining these two configurations.
- Not supported on Layer 3 port channels.

LACP Port-Channel Minimum Links and MaxBundle

A port channel aggregates similar ports to provide increased bandwidth in a single manageable interface.

The introduction of the minimum links and maxbundle feature further refines LACP port-channel operation and provides increased bandwidth in one manageable interface.

The LACP port-channel minimum links feature does the following:

- Configures the minimum number of ports that must be linked up and bundled in the LACP port channel.
- Prevents the low-bandwidth LACP port channel from becoming active.
- Causes the LACP port channel to become inactive if there are few active members ports to supply the required minimum bandwidth.

The LACP MaxBundle defines the maximum number of bundled ports allowed in a LACP port channel.

The LACP MaxBundle feature does the following:

- Defines an upper limit on the number of bundled ports in an LACP port channel.
- Allows hot-standby ports with fewer bundled ports. (For example, in an LACP port channel with four ports, you can designate two of those ports as hot-standby ports.)

Note The minimum links and maxbundle feature works only with LACP port channels. However, the device allows you to configure this feature in non-LACP port channels, but the feature is not operational.

LACP Fast Timers

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the lacp rate command to set the rate at which LACP control packets are sent to an LACP-supported interface. You can change the timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only on LACP-enabled interfaces. To configure the LACP fast time rate, see the *Configuring the LACP Fast Timer Rate* section.

High Availability

Port channels provide high availability by load balancing traffic across multiple ports. If a physical port fails, the port channel is still operational if there is an active member in the port channel. You can bundle ports from different modules and create a port channel that remains operational even if a module fails because the settings are common across the module.

Port channels support stateful and stateless restarts.

The port channel goes down if the operational ports fall below the configured minimum links number.

Note See the *Cisco Nexus High Availability and Redundancy Guide* for complete information about high-availability features.

Prerequisites for Port Channeling

Port channeling has the following prerequisites:

- You must be logged onto the device.
- All ports for a single port channel must be either Layer 2 or Layer 3 ports.
- All ports for a single port channel must meet the compatibility requirements. See the Compatibility Requirements, on page 479 section for more information about the compatibility requirements.

Guidelines and Limitations

Port channeling has the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- The LACP port-channel minimum links and maxbundle feature is not supported for host interface port channels.
- Enable LACP before you can use that feature.
- You can configure multiple port channels on a device.
- Do not put shared and dedicated ports into the same port channel. (See the *Configuring Basic Interface Parameters* chapter for information about shared and dedicated ports.)
- For Layer 2 port channels, ports with different STP port path costs can form a port channel if they are compatibly configured with each other. See the Compatibility Requirements, on page 479 section for more information about the compatibility requirements.
-
- In STP, the port-channel cost is based on the aggregated bandwidth of the port members.
- After you configure a port channel, the configuration that you apply to the port channel interface affects the port channel member ports. The configuration that you apply to the member ports affects only the member port where you apply the configuration.
- LACP does not support half-duplex mode. Half-duplex ports in LACP port channels are put in the suspended state.
- A maximum of 12 port channels can be supported by Cisco Nexus 3550-T switches system-wide.

Default Settings

The following table lists the default settings for port-channel parameters.

Table 30: Default Port-Channel Parameters

Parameters	Default
Port channel	Admin up
Load balancing method for Layer 3 interfaces	Source and destination IP address
Load balancing method for Layer 2 interfaces	Source and destination MAC address
Load balancing per module	Disabled
LACP	Disabled
Channel mode	on
LACP system priority	32768
LACP port priority	32768
Minimum links for LACP	1
Maxbundle	4

Configuring Port Channels

Note See the *Configuring Layer 3 Interfaces* chapter for information about configuring IPv4 addresses on the port-channel interface.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Creating a Port Channel

You can create a port channel before you create a channel group. The software automatically creates the associated channel group.

Note When the port channel is created before the channel group, the port channel should be configured with all of the interface attributes that the member interfaces are configured with. Use the **switchport mode trunk** {*allowed vlan vlan-id* | *native vlan-id*} command to configure the members.

This is required only when the channel group members are Layer 2 ports (switchport) and trunks (switchport mode trunk).

Note Use the **no interface port-channel** command to remove the port channel and delete the associated channel group.

Command	Purpose
no interface port-channel *channel-number* **Example:** `switch(config)# no interface port-channel 1`	Removes the port channel and deletes the associated channel group.

Before you begin

Enable LACP if you want LACP-based port channels.

Procedure

	Command or Action	Purpose

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

Step 2	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config-if)`
		Specifies the port-channel interface to configure, and enters the interface configuration mode. The range is from 1 to 4096. The Cisco NX-OS software automatically creates the channel group if it does not already exist.

Step 3	**show port-channel summary** **Example:** `switch(config-router)# show port-channel summary`	(Optional) Displays information about the port channel.

Step 4	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e1/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to create a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
```

See the Compatibility Requirements, on page 479 section for details on how the interface configuration changes when you delete the port channel.

Adding a Layer 2 Port to a Port Channel

You can add a Layer 2 port to a new channel group or to a channel group that already contains Layer 2 ports. The software creates the port channel associated with this channel group if the port channel does not already exist.

Note Use the **no channel-group** command to remove the port from the channel group.

Command	Purpose
no channel-group **Example:** `switch(config)# no channel-group`	Removes the port from the channel group.

Before you begin

Enable LACP if you want LACP-based port channels.

All Layer 2 member ports must run in full-duplex mode and at the same speed

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface that you want to add to a channel group, and enters the interface configuration mode.

Step 3	**switchport** **Example:** `switch(config)# switchport`	Configures the interface as a Layer 2 access port.

	Command or Action	Purpose
Step 4	**switchport mode trunk** **Example:** `switch(config)# switchport mode trunk`	(Optional) Configures the interface as a Layer 2 trunk port.

Step 5	**switchport trunk** {**allowed vlan** *vlan-id* | **native** *vlan-id*}
	Example:
	`switch(config)# switchport trunk native 3`
	`switch(config-if)#`
		(Optional) Configures necessary parameters for a Layer 2 trunk port.

Step 6	**channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}] **Example:** • `switch(config-if)# channel-group 5` • `switch(config-if)# channel-group 5 force`	Configures the port in a channel group and sets the mode. The channel-number range is from 1 to 4096. This command creates the port channel associated with this channel group if the port channel does not already exist. All static port-channel interfaces are set to mode **on**. You must set all LACP-enabled port-channel interfaces to **active** or **passive**. The default mode is **on**. (Optional) Forces an interface with some incompatible configurations to join the channel. The forced interface must have the same speed, duplex, and flow control settings as the channel group.
Step 7	**show interface** *type slot/port* **Example:** `switch# show interface port channel 5`	(Optional) Displays interface information.

Step 8	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e1/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 9	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to add a Layer 2 Ethernet interface 1/4 to channel group 5:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
```

```
switch(config-if)# switchport
switch(config-if)# channel-group 5
```

Adding a Layer 3 Port to a Port Channel

You can add a Layer 3 port to a new channel group or to a channel group that is already configured with Layer 3 ports. The software creates the port channel associated with this channel group if the port channel does not already exist.

If the Layer 3 port that you are adding has a configured IP address, the system removes that IP address before adding the port to the port channel. After you create a Layer 3 port channel, you can assign an IP address to the port-channel interface.

Note Use the **no channel-group** command to remove the port from the channel group. The port reverts to its original configuration. You must reconfigure the IP addresses for this port.

Command	Purpose
no channel-group **Example:** `switch(config)# no channel-group`	Removes the port from the channel group.

Before you begin

Enable LACP if you want LACP-based port channels.

Remove any IP addresses configured on the Layer 3 interface.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface that you want to add to a channel group, and enters the interface configuration mode.

Step 3	**no switchport** **Example:** `switch(config-if)# no switchport`	Configures the interface as a Layer 3 port.
Step 4	**channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}]	Configures the port in a channel group and sets the mode. The channel-number range is from 1

	Command or Action	Purpose
	Example: • `switch(config-if)#` **`channel-group 5`** • `switch(config-if)#` **`channel-group 5 force`**	to 4096. The Cisco NX-OS software creates the port channel associated with this channel group if the port channel does not already exist. (Optional) Forces an interface with some incompatible configurations to join the channel. The forced interface must have the same speed, duplex, and flow control settings as the channel group.
Step 5	**show interface** *type slot/port* **Example:** `switch#` **`show interface ethernet 1/4`**	(Optional) Displays interface information.

Step 6	**no shutdown**
	Example:
	`switch#` **`configure terminal`**
	`switch(config)#` **`int e1/1`**
	`switch(config-if)#` **`no shutdown`**
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:** `switch(config)#` **`copy running-config startup-config`**	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to add a Layer 3 Ethernet interface 1/5 to channel group 6 in on mode:

```
switch# configure terminal
switch (config)# interface ethernet 1/5
switch(config-if)# switchport
switch(config-if)# channel-group 6
```

This example shows how to create a Layer 3 port-channel interface and assign the IP address:

```
switch# configure terminal
switch (config)# interface port-channel 4
switch(config-if)# ip address 192.0.2.1/8
```

Configuring the Bandwidth and Delay for Informational Purposes

The bandwidth of the port channel is determined by the number of total active links in the channel.

You configure the bandwidth and delay on port-channel interfaces for informational purposes.

Procedure

	Command or Action	Purpose

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

Step 2	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 2`
	`switch(config-if)#`
		Specifies the port-channel interface that you want to configure, and enters the interface mode.

Step 3	**bandwidth** *value*
	Example:
	`switch(config-if)# bandwidth 60000000`
	`switch(config-if)#`
		Specifies the bandwidth, which is used for informational purposes. The range is from 1 to 3,200,000,000 kbs. The default value depends on the total active interfaces in the channel group.

Step 4	**delay** *value*
	Example:
	`switch(config-if)# delay 10000`
	`switch(config-if)#`
		Specifies the throughput delay, which is used for informational purposes. The range is from 1 to 16,777,215 tens of microseconds. The default value is 10 microseconds.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode and returns to the configuration mode.

Step 6	**show interface port-channel** *channel-number* **Example:** `switch# show interface port-channel 2`	(Optional) Displays interface information for the specified port channel.
Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to configure the informational parameters of the bandwidth and delay for port channel 5:

```
switch# configure terminal
switch (config)# interface port-channel 5
switch(config-if)# bandwidth 60000000
switch(config-if)# delay 10000
switch(config-if)#
```

Shutting Down and Restarting the Port-Channel Interface

You can shut down and restart the port-channel interface. When you shut down a port-channel interface, no traffic passes and the interface is administratively down.

Procedure

	Command or Action	Purpose

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

Step 2	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 2`
	`switch(config-if)#`
		Specifies the port-channel interface that you want to configure, and enters the interface mode.

Step 3	**shutdown**
	Example:
	`switch(config-if)# shutdown`
	`switch(config-if)#`
		Shuts down the interface. No traffic passes and the interface displays as administratively down. The default is no shutdown.
		Note Use the **no shutdown** command to open the interface.
		The interface displays as administratively up. If there are no operational problems, traffic passes. The default is no shutdown.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode and returns to the configuration mode.

Step 5	**show interface port-channel** *channel-number* **Example:** `switch(config-router)# show interface port-channel 2`	(Optional) Displays interface information for the specified port channel.

Step 6	**no shutdown**
	Example:
	`switch# configure terminal`
	`switch(config)# int e1/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:**	(Optional) Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config)# copy running-config startup-config`	

Example

This example shows how to bring up the interface for port channel 2:

```
switch# configure terminal
switch (config)# interface port-channel 2
switch(config-if)# no shutdown
```

Configuring a Port-Channel Description

You can configure a description for a port channel.

Procedure

	Command or Action	Purpose

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

Step 2	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 2`
	`switch(config-if)#`
		Specifies the port-channel interface that you want to configure, and enters the interface mode.

Step 3	**description**
	Example:
	`switch(config-if)# description engineering`
	`switch(config-if)#`
		Allows you to add a description to the port-channel interface. You can use up to 80 characters in the description. By default, the description does not display; you must configure this parameter before the description displays in the output.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode and returns to the configuration mode.

Step 5	**show interface port-channel** *channel-number* **Example:** `switch# show interface port-channel 2`	(Optional) Displays interface information for the specified port channel.
Step 6	**copy running-config startup-config** **Example:**	(Optional) Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config)# copy running-config startup-config`	

Example

This example shows how to add a description to port channel 2:

```
switch# configure terminal
switch (config)# interface port-channel 2
switch(config-if)# description engineering
```

Configuring the Speed and Duplex Settings for a Port-Channel Interface

You can configure the speed and duplex settings for a port-channel interface.

Procedure

	Command or Action	Purpose

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

Step 2	**interface port-channel** *channel-number*
	Example:
	`switch(config)# interface port-channel 2`
	`switch(config-if)#`
		Specifies the port-channel interface that you want to configure, and enters the interface mode.

Step 3	**speed** {**auto**}
	Example:
	`switch(config-if)# speed auto`
	`switch(config-if)#`
		Sets the speed for the port-channel interface. The default is auto for autonegotiation.

Step 4	**duplex** {**auto** | **full** | **half**}
	Example:
	`switch(config-if)# speed auto`
	`switch(config-if)#`
		Sets the duplex for the port-channel interface. The default is auto for autonegotiation.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode and returns to the configuration mode.

Step 6	**show interface port-channel** *channel-number* **Example:**	(Optional) Displays interface information for the specified port channel.

	Command or Action	Purpose
	`switch# show interface port-channel 2`	
Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set port channel 2 to 100 Mb/s:

```
switch# configure terminal
switch (config)# interface port-channel 2
switch(config-if)# speed 100
```

Configuring Load Balancing Using Port Channels

You can configure the load-balancing algorithm for port channels that applies to the entire device.

Note Use the **no port-channel load-balance** command to restore the default load-balancing algorithm of source-dest-mac for non-IP traffic and source-dest-ip for IP traffic.

Command	Purpose
no port-channel load-balance **Example:** `switch(config)# no port-channel load-balance`	Restores the default load-balancing algorithm.

Before you begin

Enable LACP if you want LACP-based port channels.

Procedure

	Command or Action	Purpose

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

Step 2	**port-channel load-balance** *method* {**dst ip** | **dst ip-l4port** | **dst ip-l4port-vlan** | **dst ip-vlan** | **dst l4port** | **dst mac** | **src ip** | **src ip-l4port** | **src ip-l4port-vlan** | **src ip-vlan** | **src l4port** |	Specifies the load-balancing algorithm for the device. The range depends on the device. The default for Layer 3 is **src-dst ip-l4port** for IPv4, and the default for non-IP is **src-dst mac**.

	Command or Action	Purpose
	src mac | **src-dst ip** | **src-dst ip-l4port** [**symmetric**] | **src-dst ip-l4port-vlan** | **src-dst ip-vlan** | **src-dst l4port** | **src-dst mac**} [{*all*}] [**rotate** *rotate*] **Example:** • `switch(config)# port-channel load-balance src-dst mac` `switch(config)#` • `switch(config)# no port-channel load-balance src-dst mac` `switch(config)#`	**Note** Only the following load-balancing algorithms support symmetric hashing: • src-dst ip • src-dst ip-l4port
Step 3	**show port-channel load-balance** **Example:** `switch(config-router)# show port-channel load-balance`	(Optional) Displays the port-channel load-balancing algorithm.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Enabling LACP

LACP is disabled by default; you must enable LACP before you begin LACP configuration. You cannot disable LACP while any LACP configuration is present.

LACP learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once LACP identifies correctly matched Ethernet links, it group the links into a port channel. The port channel is then added to the spanning tree as a single bridge port.

To configure LACP, you must do the following:

- Enable LACP globally by using the **feature lacp** command.
- You can use different modes for different interfaces within the same LACP-enabled port channel. You can change the mode between **active** and **passive** for an interface only if it is the only interface that is designated to the specified channel group.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**feature lacp** **Example:** `switch(config)# feature lacp`	Enables LACP on the device.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to enable LACP:

```
switch# configure terminal
switch (config)# feature lacp
```

Configuring LACP Port-Channel Port Modes

After you enable LACP, you can configure the channel mode for each individual link in the LACP port channel as **active** or **passive**. This channel configuration mode allows the link to operate with LACP.

When you configure port channels with no associated aggregation protocol, all interfaces on both sides of the link remain in the **on** channel mode.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface that you want to add to a channel group, and enters the interface configuration mode.

Step 3	**channel-group** *number* **mode** {**active** | **on** | **passive**}
	Example:
	`switch(config-if)# channel-group 5 mode active`
		Specifies the port mode for the link in a port channel. After LACP is enabled, you configure each link or the entire channel as active or passive.
		When you run port channels with no associated aggregation protocol, the port-channel mode is always on.
		The default port-channel mode is **on**.

	Command or Action	Purpose
Step 4	**show port-channel summary** **Example:** `switch(config-if)# show port-channel summary`	(Optional) Displays summary information about the port channels.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set the LACP-enabled interface to the active port-channel mode for Ethernet interface 1/4 in channel group 5:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# channel-group 5 mode active
```

Configuring LACP Port-Channel Minimum Links

You can configure the LACP minimum links feature. Although minimum links and maxbundles work only in LACP, you can enter the CLI commands for these features for non-LACP port channels, but these commands are nonoperational.

Note Use the **no lacp min-links** command to restore the default port-channel minimum links configuration.

Command	Purpose
no lacp min-links **Example:** `switch(config)# no lacp min-links`	Restores the default port-channel minimum links configuration.

Before you begin

Ensure that you are in the correct port-channel interface.

Procedure

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

	Command or Action	Purpose

	`switch# configure terminal`
	`switch(config)#`	

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 3`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 3	**lacp min-links** *number* **Example:** `switch(config-if)# lacp min-links 3`	Specifies the port-channel interface to configure the number of minimum links. The range is from 1 to 4.
Step 4	**show running-config interface port-channel** *number* **Example:** `switch(config-if)# show running-config interface port-channel 3`	(Optional) Displays the port-channel minimum links configuration.

Example

This example shows how to configure the minimum number of port-channel member interfaces to be up/active for the port-channel to be up/active:

```
switch# configure terminal
switch(config)# interface port-channel 3
switch(config-if)# lacp min-links 3
```

Configuring the LACP Port-Channel MaxBundle

You can configure the LACP maxbundle feature. Although minimum links and maxbundles work only in LACP, you can enter the CLI commands for these features for non-LACP port channels, but these commands are nonoperational.

Note Use the **no lacp max-bundle** command to restore the default port-channel max-bundle configuration.

Command	Purpose
no lacp max-bundle **Example:** `switch(config)# no lacp max-bundle`	Restores the default port-channel max-bundle configuration.

Before you begin

Ensure that you are in the correct port-channel interface.

Procedure

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# `**`interface port-channel 3`**
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 3	**lacp max-bundle** *number* **Example:** `switch(config-if)# lacp max-bundle`	Specifies the port-channel interface to configure max-bundle. The default value for the port-channel max-bundle is 4. The allowed range is from 1 to 4. **Note** Even if the default value is 4, the number of active members in a port channel is the minimum of the pc_max_links_config and pc_max_active_members that is allowed in the port channel.
Step 4	**show running-config interface port-channel** *number* **Example:** `switch(config-if)# `**`show running-config interface port-channel 3`**	(Optional) Displays the port-channel max-bundle configuration.

Example

This example shows how to configure the port channel interface max-bundle:

```
switch# configure terminal
switch(config)# interface port-channel 3
switch(config-if)# lacp max-bundle 3
```

Configuring the LACP Fast Timer Rate

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the **lacp rate** command to set the rate at which LACP control packets are sent to an LACP-supported interface. You can change the timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only on LACP-enabled interfaces.

Note We do not recommend changing the LACP timer rate. HA and SSO are not supported when the LACP fast rate timer is configured.

Before you begin

Ensure that you have enabled the LACP feature.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure and enters the interface configuration mode.

Step 3	**lacp rate fast** **Example:** `switch(config-if)# lacp rate fast`	Configures the fast rate (one second) at which LACP control packets are sent to an LACP-supported interface. To reset the timeout rate to its default, use the **no** form of the command.

Example

This example shows how to configure the LACP fast rate on Ethernet interface 1/4:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# lacp rate fast
```

This example shows how to restore the LACP default rate (30 seconds) on Ethernet interface 1/4.

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# no lacp rate fast
```

Configuring the LACP System Priority

The LACP system ID is the combination of the LACP system priority value and the MAC address.

Before you begin

Enable LACP.

Procedure

	Command or Action	Purpose

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

Step 2	**lacp system-priority** *priority* **Example:** `switch(config)# lacp system-priority 40000`	Configures the system priority for use with LACP. Valid values are from 1 through 65535, and higher numbers have a lower priority. The default value is 32768.
Step 3	**show lacp system-identifier** **Example:** `switch(config-if)# show lacp system-identifier`	(Optional) Displays the LACP system identifier.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set the LACP system priority to 2500:

```
switch# configure terminal
switch(config)# lacp system-priority 2500
```

Configuring the LACP Port Priority

When you enable LACP, you can configure each link in the LACP port channel for the port priority.

Before you begin

Enable LACP.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface that you want to add to a channel group, and enters the interface configuration mode.

Step 3	**lacp port-priority** *priority* **Example:** `switch(config-if)# lacp port-priority 40000`	Configures the port priority for use with LACP. Valid values are from 1 through 65535, and higher numbers have a lower priority. The default value is 32768.
Step 4	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to set the LACP port priority for Ethernet interface 1/4 to 40000:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# lacp port-priority 40000
```

Configuring LACP System MAC and Role

You can configure the MAC address used by the LACP for protocol exchanges and the optional role. By default, the role is primary.

This procedure is supported on the Cisco Nexus 3550-T switches.

Before you begin

LACP must be enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.

Step 2	**lacp system-mac** *mac-address* **role** *role-value*
	Example:
	`switch(config)# lacp system-mac 000a.000b.000c role primary`
	`switch(config)# lacp system-mac 000a.000b.000c role secondary`
		Specifies the MAC address to use in the LACP protocol exchanges. The role is optional. Primary is the default.

	Command or Action	Purpose
Step 3	(Optional) **show lacp system-identifier** **Example:** `switch(config)# show lacp system-identifier`	Displays the configured MAC address.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure the role of a switch as primary.

```
Switch1# sh lacp system-identifier
32768,0-b-0-b-0-b
Switch1# sh run  | grep lacp
feature lacp
lacp system-mac 000b.000b.000b role primary
```

The following example shows how to configure the role of a switch as secondary.

```
Switch2# sh lacp system-identifier
32768,0-b-0-b-0-b
Switch2# sh run | grep lacp
feature lacp
lacp system-mac 000b.000b.000b role secondary
```

Disabling LACP Graceful Convergence

By default, LACP graceful convergence is enabled. In situations where you need to support LACP interoperability with devices where the graceful failover defaults may delay the time taken for a disabled port to be brought down or cause traffic from the peer to be lost, you can disable convergence. If the downstream access switch is not a Cisco Nexus device, disable the LACP graceful convergence option.

Note The port channel has to be in the administratively down state before the command can be run.

Before you begin

Enable LACP.

Procedure

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

	Command or Action	Purpose

	`switch# configure terminal`
	`switch(config)#`	

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config-if)#`
		Specifies the port channel interface to configure and enters the interface configuration mode.

Step 3	**shutdown** **Example:** `switch(config-if) shutdown`	Administratively shuts down the port channel.
Step 4	**no lacp graceful-convergence** **Example:** `switch(config-if)# no lacp graceful-convergence`	Disables LACP graceful convergence on the port channel.
Step 5	**no shutdown** **Example:** `switch(config-if) no shutdown`	Brings the port channel administratively up.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to disable LACP graceful convergence on a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
switch(config-if)# shutdown
switch(config-if)# no lacp graceful-convergence
switch(config-if)# no shutdown
```

Reenabling LACP Graceful Convergence

If the default LACP graceful convergence is once again required, you can reenable convergence.

Procedure

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

	Command or Action	Purpose

	`switch# configure terminal`
	`switch(config)#`	

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config-if)#`
		Specifies the port channel interface to configure and enters the interface configuration mode.

Step 3	**shutdown** **Example:** `switch(config-if) shutdown`	Administratively shuts down the port channel.
Step 4	**lacp graceful-convergence** **Example:** `switch(config-if)# lacp graceful-convergence`	Enables LACP graceful convergence on the port channel.
Step 5	**no shutdown** **Example:** `switch(config-if) no shutdown`	Brings the port channel administratively up.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to enable LACP graceful convergence on a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
switch(config-if)# shutdown
switch(config-if)# lacp graceful-convergence
switch(config-if)# no shutdown
```

Disabling LACP Suspend Individual

LACP sets a port to the suspended state if it does not receive an LACP PDU from the peer. This process can cause some servers to fail to boot up as they require LACP to logically bring up the port.

Note You should only enter the **lacp suspend-individual** command on edge ports. The port channel has to be in the administratively down state before you can use this command.

Before you begin

Enable LACP.

Procedure

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config-if)#`
		Specifies the port channel interface to configure and enters the interface configuration mode.

Step 3	**shutdown** **Example:** `switch(config-if) shutdown`	Administratively shuts down the port channel.
Step 4	**no lacp suspend-individual** **Example:** `switch(config-if)# no lacp suspend-individual`	Disables LACP individual port suspension behavior on the port channel.
Step 5	**no shutdown** **Example:** `switch(config-if) no shutdown`	Brings the port channel administratively up.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to disable LACP individual port suspension on a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
switch(config-if)# shutdown
switch(config-if)# no lacp suspend-individual
switch(config-if)# no shutdown
```

Reenabling LACP Suspend Individual

You can reenable the default LACP individual port suspension.

Procedure

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config-if)#`
		Specifies the port channel interface to configure and enters the interface configuration mode.

Step 3	**shutdown** **Example:** `switch(config-if) shutdown`	Administratively shuts down the port channel.
Step 4	**lacp suspend-individual** **Example:** `switch(config-if)# lacp suspend-individual`	Enables LACP individual port suspension behavior on the port channel.
Step 5	**no shutdown** **Example:** `switch(config-if) no shutdown`	Brings the port channel administratively up.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to reenable the LACP individual port suspension on a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
switch(config-if)# shutdown
switch(config-if)# lacp suspend-individual
switch(config-if)# no shutdown
```

Configuring Delayed LACP

The delayed LACP feature enables one port channel member, the delayed LACP port, to come up first as a member of a regular port channel before LACP PDUs are received. You configure the delayed LACP feature using the **lacp mode delay**command on a port channel followed by configuring the LACP port priority on a one member port of the port channel.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface port-channel** *number*	Specifies the port channel interface to configure and enters the interface configuration mode.
Step 3	**lacp mode delay**	Enables delayed LACP. **Note** To disable delayed LACP, use the **no lacp mode delay** command. Complete the configuration of the delayed LACP by configuring the LACP port priority. See the "Configuring the LACP Port Priority" section for details. The priority of a LACP port determines the election of the delayed LACP port. The port with the lowest numerical priority is elected. When the delayed LACP feature is configured and made effective with a port channel flap, the delayed LACP port operates as a member of a regular port channel, allowing data to be exchanged between the server and switch. After receiving the first LACP PDU, the delayed LACP port transitions from a regular port member to a LACP port member. **Note** The election of the delayed LACP port is not complete or effective until the port channel flaps on the switch or at a remote server.

Example

The following example configures delayed LACP.

```
switch# config terminal
switch(config)# interface po 1
switch(config-if)# lacp mode delay
```

```
switch# config terminal
switch(config)# interface ethernet 1/1
switch(config-if)# lacp port-priority 1
switch(config-if)# channel-group 1 mode active
```

The following example disables delayed LACP.

```
switch# config terminal
switch(config)# interface po 1
switch(config-if)# no lacp mode delay
```

Verifying the Port-Channel Configuration

To display port-channel configuration information, perform one of the following tasks:

Command	Purpose
show interface port-channel *channel-number*	Displays the status of a port-channel interface.
show feature	Displays enabled features.
load- interval {**interval** *seconds* {**1** | **2** | **3**}}	Sets three different sampling intervals to bit-rate and packet-rate statistics.
show port-channel compatibility-parameters	Displays the parameters that must be the same among the member ports in order to join a port channel.
show port-channel database [**interface port-channel** *channel-number*]	Displays the aggregation state for one or more port-channel interfaces.
show port-channel load-balance	Displays the type of load balancing in use for port channels.
show port-channel summary	Displays a summary for the port-channel interfaces.
show port-channel traffic	Displays the traffic statistics for port channels.
show port-channel usage	Displays the range of used and unused channel numbers.
show lacp {**counters** [**interface port-channel** *channel-number*] | [**interface** *type/slot*] | **neighbor** [**interface port-channel** *channel-number*] | **port-channel** [**interface port-channel** *channel-number*] | **system-identifier**]]}	Displays information about LACP.
show running-config interface port-channel *channel-number*	Displays information about the running configuration of the port-channel.

Monitoring the Port-Channel Interface Configuration

Use the following commands to display port-channel interface configuration information.

Command	Purpose
clear counters interface port-channel *channel-number*	Clears the counters.
clear lacp counters [**interface port-channel** *channel-number*]	Clears the LACP counters.
load- interval {**interval** *seconds* {**1** | **2** | **3**}}	Sets three different sampling intervals to bit-rate and packet-rate statistics.
show interface counters [**module** *module*]	Displays input and output octets unicast packets, multicast packets, and broadcast packets.
show interface counters detailed [**all**]	Displays input packets, bytes, and multicast and output packets and bytes.
show interface counters errors [**module** *module*]	Displays information about the number of error packets.
show lacp counters	Displays statistics for LACP.

Example Configurations for Port Channels

This example shows how to create an LACP port channel and add two Layer 2 interfaces to that port channel:

```
switch# configure terminal
switch (config)# feature lacp
switch (config)# interface port-channel 5
switch (config-if)# interface ethernet 1/4
switch(config-if)# switchport
switch(config-if)# channel-group 5 mode active
switch(config-if)# lacp port priority 40000
switch(config-if)# interface ethernet 1/7
switch(config-if)# switchport
switch(config-if)# channel-group 5 mode
```

This example shows how to add two Layer 3 interfaces to a channel group. The Cisco NX-OS software automatically creates the port channel:

```
switch# configure terminal
switch (config)# interface ethernet 1/5
switch(config-if)# no switchport
switch(config-if)# no ip address
switch(config-if)# channel-group 6 mode active
switch (config)# interface ethernet 1/5
switch(config-if)# no switchport
switch(config-if)# no ip address
switch(config-if)# channel-group 6 mode active
switch (config)# interface port-channel 6
```

```
switch(config-if)# ip address 192.0.2.1/8
```

Related Documents

Related Topic	Document Title
System management	*Cisco Nexus 3550-T NX-OS System Management Configuration* section
Licensing	*Cisco NX-OS Licensing Guide*

CHAPTER 32

Configuring Layer 3 Interfaces

About Layer 3 Interfaces

Layer 3 interfaces forward IPv4 packets to another device using static or dynamic routing protocols. You can use Layer 3 interfaces for IP routing and inter-VLAN routing of Layer 2 traffic.

Routed Interfaces

You can configure a port as a Layer 2 interface or a Layer 3 interface. A routed interface is a physical port that can route IP traffic to another device. A routed interface is a Layer 3 interface only and does not support Layer 2 protocols, such as the Spanning Tree Protocol (STP).

All Ethernet ports are routed interfaces by default. You can change this default behavior with the CLI setup script.

Note The default mode for the Cisco Nexus® 3550-T switch interface is Layer 3.

You can assign an IP address to the port, enable routing, and assign routing protocol characteristics to this routed interface.

You can also create a Layer 3 port channel from routed interfaces. For more information about port channels, see the *Configuring Port Channels* section.

Routed interfaces support exponentially decayed rate counters. Cisco NX-OS tracks the following statistics with these averaging counters:

- Input packets/sec
- Output packets/sec

Note Layer 3 sub-interfaces are not supported in the Cisco Nexus® 3550-T 10.1(2t) release.

VLAN Interfaces

A VLAN interface, or switch virtual interface (SVI), is a virtual routed interface that connects a VLAN on the device to the Layer 3 router engine on the same device. Only one VLAN interface can be associated with a VLAN, but you need to configure a VLAN interface for a VLAN only when you want to route between VLANs or to provide IP host connectivity. When you enable VLAN interface creation, Cisco NX-OS creates a VLAN interface for the default VLAN (VLAN 1) to permit remote switch administration.

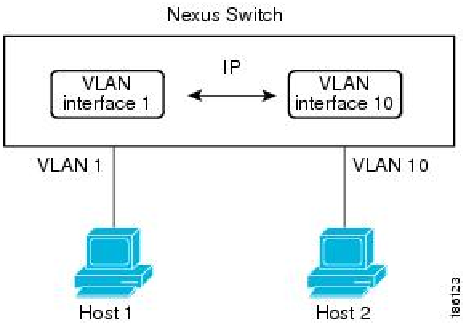
You must enable the VLAN network interface feature before you can see configure it. The system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint. See the *Cisco Nexus® 3550-T System Management Configuration* section for information on rollbacks and checkpoints.

Note You cannot delete the VLAN interface for VLAN 1.

You can route across VLAN interfaces to provide Layer 3 inter-VLAN routing by configuring a VLAN interface for each VLAN that you want to route traffic to and assigning an IP address on the VLAN interface. For more information about IP addresses and IP routing, see the *Cisco Nexus® 3550-T Unicast Routing Configuration* section.

The following figure shows two hosts connected to two VLANs on a device. You can configure VLAN interfaces for each VLAN that allows Host 1 to communicate with Host 2 using IP routing between the VLANs. VLAN 1 communicates at Layer 3 over VLAN interface 1 and VLAN 10 communicates at Layer 3 over VLAN interface 10.

Figure 28: Connecting Two VLANs with VLAN interfaces



Note In Cisco Nexus® 3550-T 10.1(2t) release, SVI interfaces are only supported in the default VRF instances.

Loopback Interfaces

A loopback interface is a virtual interface with a single endpoint that is always up. Any packet transmitted over a loopback interface is immediately received by this interface. Loopback interfaces emulate a physical interface. You can configure up to 1024 loopback interfaces, numbered 0 to 1023.

You can use loopback interfaces for performance analysis, testing, and local communications. Loopback interfaces can act as a termination address for routing protocol sessions. This loopback configuration allows routing protocol sessions to stay up even if some of the outbound interfaces are down.

Prerequisites for Layer 3 Interfaces

Layer 3 interfaces have the following prerequisites:

- You are familiar with IP addressing and basic configuration. See the *Cisco Nexus® 3550-T Unicast Routing Configuration* section for more information about IP addressing.

Guidelines and Limitations for Layer 3 Interfaces

Layer 3 interfaces have the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- The Dynamic Host Configuration Protocol (DHCP) option is not supported in ***Cisco Nexus 3550-T - 10.1(2t) release***.
- Layer 3 sub-interfaces are not supported ***Cisco Nexus 3550-T - 10.1(2t) release***.
- SVI interfaces are only supported in Default VRF instances in ***Cisco Nexus 3550-T - 10.1(2t) release***.
- MTU Check is not supported in ***Cisco Nexus 3550-T - 10.1(2t) release*** and MTU CLI's do not take effect. Control-plane adjacencies would not be formed when peering devices send packets larger than 1518 bytes.
- ***Cisco Nexus 3550-T - 10.1(2t) release*** switch does cut-through forwarding; hence there is no MTU-check implemented.

  Hardware buffering is not designed for jumbo packets and packets beyond regular MTU size 1516 is not supported.
- There is no support for VLAN packet and byte counters in ***Cisco Nexus 3550-T - 10.1(2t) release***.
- ***Cisco Nexus 3550-T - 10.1(2t) release*** release does not support any byte counters on any interface. All these counters will display as 0.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings

The following table lists the default settings for Layer 3 interface parameters.

Table 31: Default Layer 3 Interface Parameters

Parameters	Default
Admin state	Shut

Configuring Layer 3 Interfaces

Configuring a Routed Interface

You can configure any Ethernet port as a routed interface.

Procedure

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**no switchport** **Example:** `switch(config-if)# no switchport`	Configures the interface as a Layer 3 interface.
Step 4	[**ip address**] **Example:** `switch(config-if)# ip address 192.0.2.1/8`	• Configures an IP address for this interface. See the *Cisco Nexus® 3550-T Unicast Routing Configuration* section for more information about IP addresses.
Step 5	**show interfaces** **Example:** `switch(config-if)# show interfaces ethernet 1/1`	(Optional) Displays the Layer 3 interface statistics.

	Command or Action	Purpose

Step 6	**no shutdown**
	Example:
	`switch#`
	`switch(config-if)# int e1/1`
	`switch(config-if)# no shutdown`
		(Optional) Clears the errors on the interfaces where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the configuration change.

Example

- Use the **medium** command to set the interface medium to either point to point or broadcast.

Command	Purpose
switchport Example: `switch(config-if)# switchport`	Configures the interface as a Layer 2 interface and deletes any configuration specific to Layer 3 on this interface.

- This example shows how to configure a routed interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# no switchport
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```

The default setting for interfaces is routed. If you want to configure an interface for Layer 2, enter the **switchport** command. Then, if you change a Layer 2 interface to a routed interface, enter the **no switchport** command.

Configuring a VLAN Interface

You can create VLAN interfaces to provide inter-VLAN routing.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**feature interface-vlan** **Example:** `switch(config)# feature interface-vlan`	Enables VLAN interface mode.

Step 3	**interface vlan** *number*
	Example:
	`switch(config)# interface vlan 10`
	`switch(config-if)#`
		Creates a VLAN interface. The number range is from 1 to 4094.

Step 4	[**ip address** *ip-address/length*] **Example:** `switch(config-if)# ip address 192.0.2.1/8`	• Configures an IP address for this VLAN interface. See the *Cisco Nexus® 3550-T Unicast Routing Configuration* section for more information on IP addresses.
Step 5	**show interface vlan** *number* **Example:** `switch(config-if)# show interface vlan 10`	(Optional) Displays the Layer 3 interface statistics.
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

Example

This example shows how to create a VLAN interface:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# interface vlan 10
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```

Configuring a Loopback Interface

You can configure a loopback interface to create a virtual interface that is always up.

Before you begin

Ensure that the IP address of the loopback interface is unique across all routers on the network.

Procedure

	Command or Action	Purpose

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

Step 2	**interface loopback** *instance*
	Example:
	`switch(config)# interface loopback 0`
	`switch(config-if)#`
		Creates a loopback interface. The range is from 0 to 1023.

Step 3	[**ip address** *ip-address/length*] **Example:** `switch(config-if)# ip address 192.0.2.1/8`	• Configures an IP address for this interface. See the *Cisco Nexus® 3550-T Unicast Routing Configuration* section for more information about IP addresses.
Step 4	**show interface loopback** *instance* **Example:** `switch(config-if)# show interface loopback 0`	(Optional) Displays the loopback interface statistics.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves the configuration change.

Example

This example shows how to create a loopback interface:

```
switch# configure terminal
switch(config)# interface loopback 0
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```

Configuring a DHCP Client on an Interface

You can configure the DHCP client on an SVI, a management interface, or a physical Ethernet interface for IPv4 address

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **interface | mgmt** *0* **| vlan** *vlan id*	Selects a management interface.
Step 3	switch(config-if)# [**no**] [**ip | ipv4] address dhcp**	Requests the DHCP server for an IPv4 address. The **no** form of this command removes any address that was acquired.
Step 4	switch# **configure terminal**	Enters global configuration mode.

Example

This example shows how to configure the IP address of a DHCP client on an SVI:

```
switch# configure terminal
switch(config)# interface mgmt 0
switch(config-if)# ip address dhcp
```

Verifying the Layer 3 Interfaces Configuration

To display the Layer 3 configuration, perform one of the following tasks:

Command	Purpose
show interface ethernet *slot/port*	Displays the Layer 3 interface configuration, status, and counters (including the 5-minute exponentially decayed moving average of inbound and outbound packet rates).
show interface ethernet *slot/port* **brief**	Displays the Layer 3 interface operational status.
show interface ethernet *slot/port* **capabilities**	Displays the Layer 3 interface capabilities, including port type, speed, and duplex.
show interface ethernet *slot/port* **description**	Displays the Layer 3 interface description.
show interface ethernet *slot/port* **status**	Displays the Layer 3 interface administrative status, port mode, speed, and duplex.
show interface loopback *number*	Displays the loopback interface configuration, status, and counters.
show interface loopback *number* **brief**	Displays the loopback interface operational status.
show interface loopback *number* **description**	Displays the loopback interface description.
show interface loopback *number* **status**	Displays the loopback interface administrative status and protocol status.

Command	Purpose
show interface vlan *number*	Displays the VLAN interface configuration, status, and counters.
show interface vlan *number* **brief**	Displays the VLAN interface operational status.
show interface vlan *number* **description**	Displays the VLAN interface description.
show interface vlan *number* **status**	Displays the VLAN interface administrative status and protocol status.

Monitoring the Layer 3 Interfaces

Use the following commands to display Layer 3 statistics:

Command	Purpose
load- interval {**interval** *seconds* {**1** | **2** | **3**}}	Cisco Nexus® 3550-T devices set three different sampling intervals to packet-rate statistics. The range for VLAN network interface is 60 to 300 seconds, and the range for Layer interfaces is 30 to 300 seconds.
show interface ethernet *slot/port* **counters**	Displays the Layer 3 interface statistics (unicast, multicast, and broadcast).
show interface ethernet *slot/port* **counters brief**	Displays the Layer 3 interface input and output counters.
show interface ethernet errors *slot/port* **detailed** [**all**]	Displays the Layer 3 interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).
show interface ethernet errors *slot/port* **counters errors**	Displays the Layer 3 interface input and output errors.
show interface ethernet errors *slot/port* **counters snmp**	Displays the Layer 3 interface counters reported by SNMP MIBs.
show interface loopback *number* **counters**	Displays the loopback interface input and output counters (unicast, multicast, and broadcast).
show interface loopback *number* **detailed** [**all**]	Displays the loopback interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).
show interface loopback *number* **counters errors**	Displays the loopback interface input and output errors.

Configuration Examples for Layer 3 Interfaces

This example shows how to configure Ethernet subinterfaces:

```
interface ethernet 1/1.10
description Layer 3
ip address 192.0.2.1/8
```

This example shows how to configure a loopback interface:

```
interface loopback 3
ip address 192.0.2.2/32
```

Related Documents

Related Documents	Document Title
IP	*Cisco Nexus® 3550-T Unicast Routing Configuration* section
VLANs	*Cisco Nexus® 3550-T Layer 2 Switching Configuration* section