EtherWAN

Hardened Managed 24-port 10/100/1000BASE-T (8-port combo SFP) and 4-port 10G SFP+ Layer 3 Switch

User's Guide - CLI

FastFind Links

Disclaimer of Liability

The information contained in this document is subject to change without notice. EtherWAN is not liable for any errors or omissions contained herein or for resulting damage in connection with the information provided in this manual.

All other trademarks are property of their respective owners.

Warranty

For details on the EtherWAN warranty replacement policy, please visit our web site at:

www.etherwan.com

Products Supported by this Manual:

EG99000

Preface

Audience

This guide is designed for the person who installs, configures, deploys, and maintains the Ethernet network. This document assumes the reader has moderate hardware, computer, and Internet skills.

Document Revision Level

This section provides a history of the revision changes to this document.

Revision	Document Version	Date	Description
A	Version 1	12/11/2018	

Changes in this Revision

This is first version of this document.

Document Conventions

This guide uses the following conventions to draw your attention to certain information.

Safety and Warnings

This guide uses the following symbols to draw your attention to certain information.

Symbol	Meaning	Description
	Note	Notes emphasize or supplement important points of the main text.
	Tip	Tips provide helpful information, guidelines, or suggestions for performing tasks more effectively.
	Warning	Warnings indicate that failure to take a specified action could result in damage to the device, or could result in serious bodily injury.

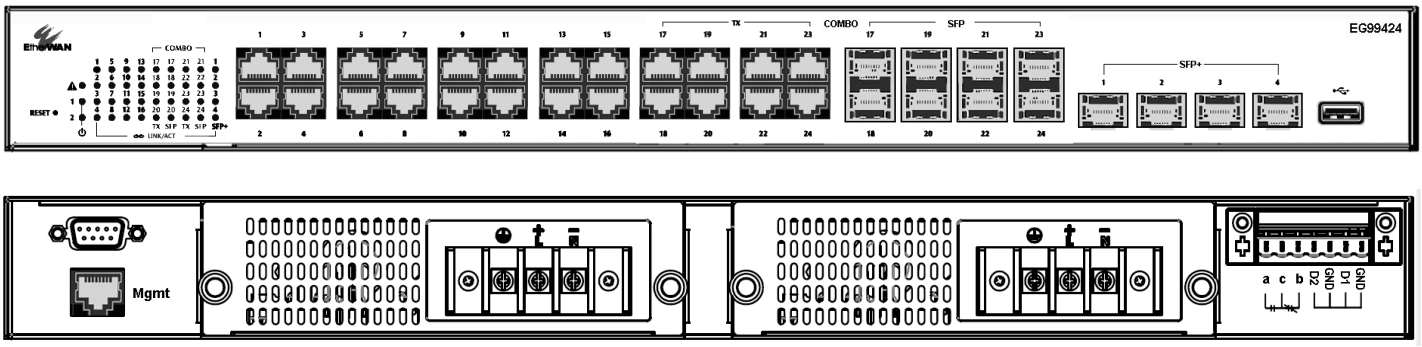
Contents

1 Introduction

EtherWAN's EG99000 is a gigabit Layer 3 switch designed for high bandwidth uplink or interconnection. With full wire speed switching capability, the EG99000 provides IP routing and switching across VLANs and subnetworks with no compromise in performance. The EG99000 supports comprehensive internetwork IP routings including static route, RIP v1 & v2, and OSPF v2 for IPv4. All these routing protocols can operate simultaneously with redistributions to each other and route control tools, including IP prefix-list and route-map.

In addition to supporting Layer 3 features, the EG99000 supports full sets of EtherWAN Layer 2 features such as port security, IGMP snooping, port-based VLAN, GARP protocols, link aggregation, access control lists and STP/RSTP/MSTP. Besides in-band management via web browser, Telnet, SSH and SNMP, the EG99000 supports out-band management via an RJ-45 port and an RS-232 console interface.

The EG99000 Series provides high reliability and nonstop operation in harsh environments where temperatures range from -40° to 75°C (-40° to 167°F), as well as in areas with high electromagnetic interference (EMI). The EG99000 is also equipped with sophisticated network and system failure recovery features including VRRP, and dual redundant power supplies to minimize the chance of network or system downtime. This makes it an ideal choice for both industrial and mission critical applications where sustained connectivity is crucial.The switch is shipped ready for use.



Unpacking and Installation

Unpacking

Unpack the items and confirm that no items are missing or damaged. Your package should include:

- One EG99000 switch
- One CD containing this user's guide
- One quick start guide

If any item is damaged or missing, notify your authorized EtherWAN representative. Keep the carton, including the original packing material, in case you need to store the product or return it.

Installing the Switch

Installation is bracket-mount. Use the enclosed screws and brackets to mount the switch in an open or enclosed rack.

- Select a power source within 6 feet (1.8 meters).
- Choose a dry area with ambient temperature between -40 and 75ºC (-40 and 167ºF).
- Be sure there is adequate airflow.

Connecting to the Data Ports

The EG99000 has the following ports:

- 24 x 10/100/1000 Mbps copper ports
- 8 x Gigabit combo ports (RJ-45 & SPF)
- 4 x 1/10G SPF+ slots
- 1 x RJ-45 Management port
- 1 x USB port

1 Gbps Combo/SFP Ports

Ports 17 – 24 are combo ports, and have two physical interfaces for each port. These ports can be used as either 10/100/1000BASE-TX on the left section or 1000BASE-FX on the right section. These ports operate in "either/or" fashion, i.e., connecting to fiber port 17 will render copper port 17 inoperable.

SPF+ Slots

SPF transceivers can be installed directly into right-side ports 17 – 24 and SPF+ ports 1 – 4. Ensure that the same type of transceiver is used at both ends of the link and that the correct type of fiber cable is used.

Connecting Power

Terminal Block

If your EG99000 comes with AC power cables, connect the cables into the power modules at the back of the switch. If your switch comes with a DC or AC terminal block (no cable), then connect the switch to a suitable power supply using 12 to 24 AWG wire. Redundant power supply is supported. However, only one power input is required to operate the switch. Input voltage is 48 VDC or 100 – 240 VAC, depending on model.

Relay Output Alarm

The switch provides one dry contact for signaling of a user-defined power or port failure. The alarm relay default is “open” and forms a closed circuit when the event occurs. The relay output can be connected to an alarm signaling device, and supports both normal open and normal closed. Relay output current is 30VDC / 1A.

Connecting to the Management Interface

Serial - Connect to the switch console by connecting the DB-9 cable to the console port of the switch and to the serial port of the computer running a terminal emulation application (such as HyperTerminal or Putty).

Configuration settings of the terminal-emulation program: Baud rate: 115,200bps, Data bits: 8, Parity: none, Stop bit: 1, Flow control: none.

TCP/IP - Connect to the switch via the RJ-45 Management port located on the rear panel below the serial port, or via one of the Ethernet ports on the front panel. Use a terminal emulation program and connect to IP address **192.168.2.10**. You can also connect to the switch through any of the RJ-45 ports on the front panel, IP address is **192.168.1.10** (VLAN1.1).

The default login name is “**root**,” **no password**.

Alternate (Backup) Firmware

Under certain circumstances, such as when there is a loss of power during an upgrade, the firmware build on the switch can become unstable. There are two firmware images stored on the switch: Active and Alternate. If the Active firmware image becomes unstable, the switch will automatically boot from the Alternate image on the next boot.

2 Command Line Interface Conventions and Usage

This manual describes accessing the EG99000 by using Telnet, SSH, or serial ports to configure the switch, using the Command Line Interface (CLI).

Navigating the CLI Hierarchy

The CLI is organized into a hierarchy of command modes. The basic modes are User exec mode, Privileged exec mode, and Global configuration mode. There are also other modes, specific to certain configurations. Each mode has its own group of commands for a specific purpose. Below are the CLI commands needed to enter a specific mode:

```
switch_a> ← User exec mode
switch_a>enable
switch_a# ← Privileged exec mode
switch_a#configure terminal
switch_a(config) ← Global configuration mode
switch_a(config) spanning-tree mst configuration
switch_a(config-mst)# ← MSTP configuration mode

switch_a(config)# line console 0
switch_a(config-line)# ← Line configuration mode

switch_a(config)# interface ge1
switch_a(config-if)# ← Interface configuration mode

switch_a(config)#vlan database
switch_a(config-vlan)# ← VLAN database configuration mode

switch_a(config)# router ospf
switch_a(config-router)# ← Router configuration mode
```

In any mode, the **exit** command will leave the current mode and enter the previous higher-order mode:

Example:

```
switch_a(config-line)# exit
switch_a(config)#
```

Saving a Configuration from the CLI

Command: **write**

Example:

```
switch_a>enable
switch_a#write
Building configuration.....
[OK]
switch_a#>
```

Command: **copy running config startup-config**

Example:

```
switch_a>enable
switch_a#copy running-config startup-config
Building configuration.....
[OK]
switch_a#>
```

CLI Keyboard Shortcuts

Ctrl + a: place cursor at the beginning of a line

Ctrl + b: backspace one character

Ctrl + d: delete one character

Ctrl + e: place cursor at the end of the line

Ctrl + f: move cursor forward one character

Ctrl + k: delete from the current position to the end of the line

Ctrl + l: redraw the command line

Ctrl + n: display the next line in the history

Ctrl + p: display the previous line in the history

Ctrl + u: delete entire line and place cursor at start of prompt

Ctrl + w: delete one word back

Command Syntax

The following symbols are used to describe the values and arguments for command entries in the CLI.

`Monospaced courier font`	Command line example.
<angle brackets>	Variable or value that must be specified.
[square brackets]	Optional parameters or arguments.
optionA | optionB	Vertical bar. Separates multiple exclusive items in a list of options.
{braces}	Indicate optional values or arguments, where one must be selected
(parentheses)	Indicates at least one or more of the values or arguments in the preceding syntax enclosed by braces must be specified.

Variable Placeholders

The command syntax uses the following tokens to represent command line variables for which you supply a value:

Token	Description
`WORD`	A contiguous text string (excluding spaces), such as IFNAME for the name of an interface
`LINE`	A text string, including spaces; no other parameters can follow this parameter
`A.B.C.D`	IPv4 address
`A.B.C.D/M`	IPv4 address and mask/prefix
`HH:MM:SS`	Time format
`AA:NN`	BGP community value
`XX:XX:XX:XX:XX:XX`	MAC address

`<1-5>`
`<1-65535>`
`<0-2147483647>`
`<0-4294967295>`
	Numeric range

Command Help

Help information is available from the command line by entering the following commands:

- **help** *command_name* Shows help for the specific command.
- **help ?** Shows commands for which there is help.
- *command_name* **?** Shows a list of arguments available.
- *string***?** (no space) Lists the possible commands that start with the string.

Command Abbreviations

The CLI accepts abbreviations that uniquely identify a keyword in commands.

Example:

```
sh in ge1
```

is is the same as

```
show interface ge1
```

3 System Commands

Terminal Line Commands

exec-timeout	
Purpose	Set the interval from last user input for system to log out
Command Mode	Line configuration
Syntax	[no] exec-timeout (<0-35791> | <0-2147483>)
Parameters	minutes, seconds
Example usage	`switch_a(config-line)#exec-timeout 20`

login	
Purpose	Enable password checking
Command Mode	Line configuration
Syntax	[no] login [local]
Parameters	Local password checking
Example usage	`switch_a(config-line)#`

privilege	
Purpose	Change privilege level for line
Command Mode	Line configuration
Syntax	[no] privilege level <1-15, 16>
Parameters	Default privilege levels are 1 – 15, max is 16
Example usage	`switch_a(config-line)#`

Basic System Configuration

show running-config	
Purpose	Display the current operating configuration, hardware and firmware versions, hostname data, etc..
Command Mode	Privilege Exec
Syntax	show running-config show running-config dns show hosts show hostname show hardware show version
Parameters	none
Example usage	`switch_a(config)# show running-config`

Ip address	
Purpose	Set the IP address for an interface.
Command Mode	Interface Configuration
Syntax	[no] ip address A.B.C.D [no] ip address A.B.C.D/M [no] ip address A.B.C.D DHCP
Parameters	A.B.C.D: IPv4 address format A.B.C.D: IPv4 address format with mask M DHCP: Dynamic Host Configuration Protocol
Example usage	`switch_a(config-if)# ip address 192.168.1.100/24`

banner	
Purpose	Display the banner motive of the day upon login.
Command Mode	Global Configuration
Syntax	banner motd default no banner motd banner motd line <STRING>
Parameters	none
Example usage	`switch_a# banner motd line Howdy`

hostname	
Purpose	Set the name of the switch
Command Mode	Global configuration
Syntax	[no] hostname <name>
Parameters	One word (use dash or underscore to separate), 1 – 64 characters
Example usage	`switch_a(config)#hostname test_switch` `test_switch(config)#`

enable password	
Purpose	Specify a password for the privilege level. Password can be an alpha-numeric string up to 80-characters, including spaces. The string cannot begin with a number.
Command Mode	Global configuration
Syntax	[no] enable password (8 | <password>)
Parameters	8 Specify that a hidden password will follow. line Specify the hidden enable password string.
Example usage	`switch_a(config)#enable password xyzzy`

ip default-gateway	
Purpose	Enable/disable a default gateway - specify the default router or next hop where IP datagrams will be forwarded if no routes are found.
Command Mode	Global configuration
Syntax	[no] ip default-gateway A.B.C.D
Parameters	none
Example usage	`switch_a(config)# ip default-gateway 10.10.10.10`

ip http server	
Purpose	To enable or disable HTTP or HTTPS
Command Mode	Global configuration
Syntax	[no] ip http server [no] ip http secure-server
Parameters	none
Example usage	`switch_a(config)# ip http secure-server`

feature telnet	
Purpose	Enable/disable telnet access to the switch
Command Mode	Global configuration
Syntax	[no] feature telnet
Parameters	none
Example usage	`switch_a(config)#feature telnet`
NOTE	If using Telnet to run the CLI Command that disables telnet, you will lose your connection.

feature ssh	
Purpose	Enable/disable SSH access to the switch
Command Mode	Global configuration
Syntax	[no] feature ssh
Parameters	none
Example usage	`switch_a(config)#feature ssh`
NOTE	If using SSH to run the CLI Command that disables SSH, you will lose your connection.

install config-file	
Purpose	Load a configuration from a TFTP server or USB flash drive
Command Mode	Privileged exec
Syntax	install config-file usb://(path/)<filename> install config-file tftp://A.B.C.D(:port)(/path)/<filename>
Parameters	IP address and port of tftp server
Example usage	`switch_a# install config-file tftp://10.10.10.10/EG99000_backup`

ip domain-list	
Purpose	Define a list of default domain names used to complete unqualified host names. Each domain in the list is to be tried in turn. The ip domain-list command is similar to the **ip domain-name** command, except that with the ip domain-list command you can define a list of domains, each to be tried in turn. If there is no domain list, the default domain name specified with the ip domain-name command is used. If there is a domain list, the default domain name is not used.
Command Mode	Global Configuration
Syntax	ip domain-list DOMAIN-NAME no ip domain-list DOMAIN-NAME
Parameters	DOMAIN-NAME: Domain name, e.g. mycompany.com
Example usage	`switch_a(config)# ip domain-list compa.com` `switch_a(config)# ip domain-list compbb.com`

ip domain-lookup	
Purpose	Enable DNS hostname-to-address translation
Command Mode	Global Configuration
Syntax	[no] ip domain-lookup
Parameters	None
Example usage	`switch_a(config)# ip domain-lookup`

ip domain-name	
Purpose	Set the detault domain name used to complete unqualified host names (names without a dotted decimal domain name).
Command Mode	Global Configuration
Syntax	ip domain-name DOMAIN-NAME no ip domain-name DOMAIN-NAME
Parameters	DOMAIN-NAME: Domain name, e.g. mycompany.com
Example usage	`switch_a(config)# ip domain-name chaon.com`

ip host	
Purpose	Define 1-2 static hostname-to-address mappings in DNS.
Command Mode	Global Configuration
Syntax	ip host WORD A.B.C.D ip host WORD (A.B.C.D) (A.B.C.D) no ip host WORD A.B.C.D no ip host WORD A.B.C.D A.B.C.D
Parameters	WORD: Hostname, such as mycompany.com A.B.C.D: IPv4 address of the host
Example usage	`switch_a(config)# ip host grooscompany.com 123.70.0.23 123.70.0.24`

ip name-server	
Purpose	Add 1-3 DNS server addresses that are used to translate hostnames to IP addresses.
Command Mode	Global Configuration
Syntax	ip name-server A.B.C.D ip name-server (A.B.C.D) (A.B.C.D) ip name-server (A.B.C.D) (A.B.C.D) (A.B.C.D) no ip name-server A.B.C.D no ip name-server A.B.C.D A.B.C.D no ip name-server A.B.C.D A.B.C.D A.B.C.D
Parameters	A.B.C.D: IPv4 address of the name server
Example usage	`switch_a(config)# ip name-server 123.70.0.23 123.70.0.24`

service auto-config enable	
Purpose	Enable auto save of configuration, and set interval
Command Mode	Global Configuration
Syntax	[no] service auto-config enable service auto-config interval <*number*>
Parameters	Number is interval time in seconds, <5-65535>
Example usage	`switch_a(config)#service auto-config enable` `switch_a(config)#service auto-config interval 10`

show firmware	
Purpose	Display the current primary and alternate firmware versions.
Command Mode	Privileged Exec
Syntax	show firmware
Parameters	none
Example usage	`switch_a# show firmware`

install image	
Purpose	Load a firmware image from a TFTP server
Command Mode	Privileged exec
Syntax	install image <A.B.C.D> FILENAME [reload]
Parameters	IP address of tftp server and filename of firmware image to load **reload** option will reboot the switch after the firmware is updated
Example usage	`switch_a# install image tftp://10.10.10.10 flash.tgz`
NOTE	Depending on the firmware being loaded, the extension may not be .tgz. The Switch does not use the extension to validate the firmware file.

reload	
Purpose	Reboot the switch
Command Mode	Privileged exec
Syntax	reload
Parameters	none
Example usage	`switch_a#reload`

logout	
Purpose	Log out from the switch
Command Mode	User Exec Mode or Privileged Exec Mode
Syntax	logout
Parameters	none
Example usage	`switch_a#logout`

restore default	
Purpose	Restore switch to default settings and reboot
Command Mode	Privileged exec
Syntax	restore default
Parameters	none
Example usage	`switch_a#restore default`

reset log file	
Purpose	Reset the current, open log file.
Command Mode	Privileged exec
Syntax	reset log file
Parameters	none
Example usage	`switch_a# reset log file`

write config-file	
Purpose	Save a configuration from a TFTP server or USB flash drive
Command Mode	Privileged exec
Syntax	write config-file [usb://<path/><filename> | tftp://server<:port></path>/<filename>]
Parameters	Server: A.B.C.D IP address of tftp server
Example usage	`switch_a# write config-file tftp://10.10.10.10/image.tgz`

4 Diagnostic Commands

System and Settings Information

show system-log	
Purpose	Shows the system log
Command Mode	User Exec or Privileged Exec
Syntax	show system-log
Parameters	none
Example usage	`switch_a# show system-log`

show cpu-usage	
Purpose	Shows current and max CPU utilization
Command Mode	Privileged Exec
Syntax	show cpu-usage
Parameters	none
Example usage	`switch_a# show cpu-usage`

show memory-usage	
Purpose	Shows memory utilization
Command Mode	Privileged Exec
Syntax	show memory-usage
Parameters	none
Example usage	`switch_a# show memory-usage`

show system-log	
Purpose	Show the system log
Command Mode	Privilegd Exec
Syntax	show system-log
Parameters	none
Example usage	`switch_a(config)#show system-log`

show rmon	
Purpose	Shows rmon data
Command Mode	Privileged Exec
Syntax	show rmon [alarm | event | history | statistics]
Parameters	alarm: RMON alarm table, event: event table, history, statistics
Example usage	`switch_a# show rmon alarm`

remote-log	
Purpose	Configure remote logging
Command Mode	Global config
Syntax	[no] remote-log enable remote-log add <A.B.C.D> remote-log del [A.B.C.D | all]
Parameters	Ip address of syslog server
Example usage	`switch_a(config)#remote-log enable` `switch_a(config)#remote-log add 192.168.1.100`

Alarm Configuration

show alarm, show alarm-trigger	
Purpose	Shows alarm information and settings for link down, power failure, temperature, and SFP transceiver
Command Mode	Privileged Exec
Syntax	show alarm show alarm-trigger link show alarm-trigger power show alarm-trigger temper show alarm-trigger sfp <temper | vcc | tx-bias | tx-pow | rx-pow> <major | minor>
Parameters	Temper: temperature, vcc: voltage, tx-bias: TX bias current, tx,rx-pow: TX, RX power major/minor: severity
Example usage	`switch_a# show alarm-trigger link`

alarm-trigger	
Purpose	Set alarms for link down, power failure, temperature, and SFP transceiver
Command Mode	Global config
Syntax	[no] alarm-trigger link <IFNAME> [no] alarm-trigger power <LEVEL> [no] alarm-trigger temper [no] alarm-trigger sfp <IFNAME> <temper | vcc | tx-bias | tx-pow | rx-pow> <major | minor>
Parameters	LEVEL is a power input (1-2), Temper: temperature, vcc: voltage, tx-bias: TX bias current, tx,rx-pow: TX, RX power, major/minor: alarm severity
Example usage	Set alarms to trigger if port ge1 goes down, power input 2 fails, an excessive temperature is reached, and signal an alarm with major severity if there is a voltage error on xe1.

	`switch(config)#alarm-trigger link ge10`
	`switch(config)#alarm-trigger power 2`
	`switch(config)#alarm-trigger temper`
	`switch(config)#alarm-trigger sfp xe1 vcc major`

DDM (Digital Diagnostics Monitoring) Configuration

show	
Purpose	Show alarm and digital input configuration and settings.
Command Mode	Privileged exec
Syntax	show temperature show sfp-alarm-trigger <IFNAME> show threshold sfp <IFNAME> show digital-input
Parameters	IFNAME: Interface name
Example usage	`switch_a# show sfp-alarm-trigger xe1`

threshold sfp	
Purpose	Set threshold values for Digital Diagnostics Monitoring (DDM) alarms on SFP modules. The SFP module must support DDM.
Command Mode	Global config
Syntax	threshold sfp IFNAME temper <high-major | low-major | high-minor | low-minor> <LEVEL> threshold sfp IFNAME vcc <high-major | low-major | high-minor | low-minor> <LEVEL> threshold sfp IFNAME tx-bias <high-major | low-major | high-minor | low-minor> <LEVEL> threshold sfp IFNAME tx-pow < high-major | low-major | high-minor | low-minor > <LEVEL> threshold sfp IFNAME rx-pow < high-major | low-major | high-minor | low-minor > <LEVEL>
Parameters	temper: Temperature threshold in Celsius <-128 - 128> vcc: Supply voltage in volts < 0 - 6.55> tx-bias: TX Bias Current threshold in milliamps <0 - 131> tx-pow: TX power threshold in dBm <-40.0 - 8.16> rx-pow: RX power threshold in dBm <-40.0 - 8.16> IFNAME: Interface name LEVEL: Numerical threshold value
Example usage	`switch(config)#threshold sfp xe1`

threshold temperature	
Purpose	Set high/low threshold value for internal temperature alarm.
Command Mode	Global config
Syntax	threshold temperature <high | low> <LEVEL>
Parameters	high: Trigger alarm when temperature is above LEVEL low: Trigger alarm when temperature is below LEVEL LEVEL: Temperature in Celsius, <-256 – 256>
Example usage	`switch(config)#threshold temperature high 150`

5 Port Commands

Port Configuration

show interface	
Purpose	Display the port status for a port, the description, port statistics, and modes of the Layer 2 interfaces
Command Mode	Privileged Exec
Syntax	show interface <IFNAME> show interface description <IFNAME> show interface statistics <IFNAME> show interface switchport bridge <GROUP NUMBER>
Parameters	IFNAME: interface name, GROUP NUMBER: the bridge group number
Example usage	`switch_a# show interface ge5`

show running-config interface	
Purpose	Display configuration for an interface.
Command Mode	Privileged Exec
Syntax	show running-config interface <IFNAME>
Parameters	IFNAME: interface name
Example usage	`switch_a# show running-config interface ge5`

description	
Purpose	Provide a custom description for a port
Command Mode	Interface Configuration
Syntax	[no] description <DESCRIPTION TEXT>
Parameters	DESCRIPTION TEXT: 1 – 80 characters, can use spaces
Example usage	`switch(config-if)#description second floor printer`

shutdown	
Purpose	Enable or disable a port
Command Mode	Interface Configuration
Syntax	[no] shutdown
Parameters	None
Example usage	`switch(config-if)#shutdown`

bandwidth	
Purpose	Set the maximum port speed for a port
Command Mode	Interface Configuration
Syntax	[no] bandwidth <1-10000000000(UNIT)>
Parameters	UNIT: <1-10000000> k for 1 to 10000000 kilobits <1-10000> m for 1 to 10000 megabits <1-10> g for 1 to 10 gigabits <1-10000000000.0> for 1 to 10000000000 bits
Example usage	`switch(config-if)#bandwidth 1000000k`

duplex	
Purpose	Set duplex to interface
Command Mode	Interface Configuration
Syntax	duplex <auto | full | half> no duplex
Parameters	Auto: auto-negotiate, full: full-duplex, half: half-duplex
Example usage	`switch(config-if)#duplex full`

flowcontrol	
Purpose	Set IEEE 802.3x Flow Control on a port
Command Mode	Interface Configuration
Syntax	no flowcontrol flowcontrol both flowcontrol receive <on / off> flowcontrol send <on / off> flowcontrol wmpause <0-255> wmcancel <0-255>
Parameters	**both**: flow control on send and receive **receive**: flow control on receive **send**: flow control on send **wmpause**: watermark pause **wmcancel**: watermark cancel
Example usage	`switch(config-if)#flowcontrol both`

show mirror	
Purpose	Show all port mirroring, show port mirroring for specific interface
Command Mode	Privileged Exec
Syntax	show port mirror [interface <IFNAME>]
Parameters	IFNAME: interface name
Example usage	`switch_a# show port mirror interface ge5`

mirror interface	
Purpose	Configure a port for port mirroring
Command Mode	Interface Configuration
Syntax	[no] mirror interface <IFNAME> direction <both | transmit | receive>
Parameters	**IFNAME:** interface name **direction both:** mirror traffic in both directions **direction receive:** mirror received traffic **direction transmit:** mirror transmitted traffic.
Example usage	`switch(config-if)#mirror interface ge2 direction transmit`
Note	This command run must separately for each source port. Port mirroring can only be performed on the same type of interfaces, e.g., only a switchport interface can mirror a switchport interface. Issuing a switchport command on a port where mirroring is enabled will remove port mirroring on that interface.

rate-control	
Purpose	Set a port rate control
Command Mode	Interface Configuration
Syntax	rate-control <ingress | egress> value <RATE>
Parameters	RATE: kbps <1-1000000>
Example usage	`switch_a(config-if)#rate-control ingress value 100000`

no switchport	
Purpose	Set the interface to Layer 3 (Routed port)
Command Mode	Interface Configuration
Syntax	no switchport
Parameters	None
Example usage	`switch_a(config-if)#no switchport`

MAC Table

show mac	
Purpose	Display MAC address information, including lists, tables, groups, and notifications
Command Mode	Privileged Exec
Syntax	show mac show mac access-lists show mac address-table show mac-access-group show mac-notification
Parameters	none
Example usage	`switch_a# show mac-access-group`

ageing-time	
Purpose	Set the amount of time that a networked device's MAC address will persist in the switch's memory before being removed
Command Mode	Global Configuration
Syntax	bridge 1 ageing-time (TIME)
Parameters	**TIME**: in seconds <10-1000000>
Example usage	`switch_a(config)# bridge 1 ageing-time 5000`

bridge acquire	
Purpose	Enable dynamic learning of mac addresses (enabled by default)
Command Mode	Global Configuration
Syntax	[no] bridge <1-32> acquire
Parameters	**<1-32>**: the bridge group ID
Example usage	`switch_a(config)# bridge 3 acquire`

clear mac address-table	
Purpose	Clear the filtering database for the default bridge. Command options are: • clear the filtering database • clear all filtering database entries configured through CLI (static) • clear all multicast filtering database entries • clear all multicast filtering database entries for a given VLAN or interface • clear all static or multicast database entries based on a mac address.
Command Mode	Privileged Exec
Syntax	clear mac address-table dynamic clear mac address-table dynamic bridge <1-32> clear mac address-table dynamic (address MACADDR | interface IFNAME (instance INST |) | vlan VID) clear mac address-table dynamic (address MACADDR | interface IFNAME (instance INST |) | vlan VID) bridge <1-32> clear mac address-table (dynamic | static | multicast) cvlan VID clear mac address-table (dynamic | static | multicast) cvlan VID svlan VID clear mac address-table (dynamic|static|multicast) cvlan VID svlan VID bridge <1-32> clear mac address-table (static | multicast) clear mac address-table (static | multicast) bridge <1-32> clear mac address-table (static | multicast) (address MACADDR | interface IFNAME | vlan VID) clear mac address-table (static|multicast) (address MACADDR | interface IFNAME | vlan VID) bridge <1-32>
Parameters	**dynamic**: Clears all dynamic entries. **multicast**: Clears all multicast filtering database entries. **static**: Clears all entries configured through management. **address**: Clear the specified MAC Address. **MACADDR**: xxxx.xxxx.xxxx format **IFNAME**: Interface name **bridge**: Clears the bridge group ID <1-32> **cvlan**: Clears all MAC address for the specified CVLAN <1-4094>. **svlan**: Clears all mac address for the specified SVLAN <1-4094>. **interface**: Clears all MAC address for the specified interface. **bridge**: Clears the bridge group ID. <1-32>. **instance**: Clears MSTP instance ID. Range is <1-63>. **vlan**: Clears all MAC address for the specified VLAN. Range is 1-4094.

Example usage	Clear all filtering database entries configured through the CLI:
	`switch_a#clear mac address-table static`
	Clear multicast filtering database entries:
	`switch_a#clear mac address-table multicast`
	Clear all filtering database entries for a given interface:

	`switch_a#clear mac address-table static interface eth0`
	Clear multicast filtering database entries for a given VLAN:
	`switch_a#clear mac address-table multicast vlan 2`
	Clear static filtering database entries for a given MAC address:
	`switch_a#clear mac address-table static address 0202.0202.0202`
	Clear all filtering database entries configured through CLI:
	`switch_a#clear mac address-table static bridge 1`
	Clear multicast filtering database entries:
	`switch_a#clear mac address-table multicast bridge 1`
	Clear all filtering database entries for a given interface:
	`switch_a#clear mac address-table static interface eth0 bridge 1`
	Clear multicast filtering database entries for a given VLAN.
	`switch_a#clear mac address-table multicast vlan 2 bridge 1`
	Clear static filtering database entries for a given MAC address:
	`switch_a#clear mac address-table static address 0202.0202.0202 bridge 1`
	Clear all filtering database entries learned through bridge operation for a given MAC address.
	`switch_a#clear mac address-table dynamic address 0202.0202.0202`

bridge max-age	
Purpose	Set the maximum age for a bridge
Command Mode	Global Configuration
Syntax	[no] bridge <1-32> max-age <6-40>
Parameters	**<1-32>**:: Bridge group ID **<6-40>**: The maximum time in seconds, to listen for the root bridge
Example usage	`switch_a(config)# bridge 2 max-age 12`
Note	Maximum age is the max time in seconds for which (if a bridge is the root bridge) a message is considered valid. The value of maximum age should be greater than twice the value of hello time plus 1, but less than twice the value of forward delay minus 1. The allowable range for max-age is 6-40 seconds. Configure this value sufficiently high, so that a frame generated by root can be propagated to leaf nodes without exceeding the maximum age.

Static MAC Entry

mac-address-table	
Purpose	Configure the static forwarding table entry for the default bridge. Use the no parameter to remove the entry for the default bridge.
Command Mode	Global Configuration
Syntax	mac-address-table static MAC (forward | discard) IFNAME mac-address-table static MAC (forward | discard) IFNAME vlan <2-4094> mac-address-table static MAC (forward | discard) IFNAME vlan <2-4094> svlan <2-4094> no mac-address-table static MAC (forward | discard) IFNAME no mac-address-table static MAC (forward | discard) IFNAME vlan <2-4094> no mac-address-table static MAC (forward | discard) IFNAME vlan <2-4094> svlan <2-4094>
Parameters	**static**: Configure a static address **MAC**: Media Access Control address in HHHH.HHHH.HHHH format. **forward**: Forward matching frames. **discard**: Discard matching frames. **IFNAME**: Interface on which the frame comes out. **vlan**: Identity of the VLAN in range of <2-4094>. **svlan**: Identity of the SVLAN in range of <2-4094>.
Example usage	`switch_a(config)# mac-address-table static 2222.2222.2222 forward ge5`

bridge address	
Purpose	Statically configure a bridge entry to forward or discard matching frames from matching MAC addresses.
Command Mode	Global Configuration
Syntax	[no] bridge <1-32> address MAC discard IFNAME [no] bridge <1-32> address MAC discard IFNAME vlan <2-4094> [no] bridge <1-32> address MAC discard IFNAME vlan <2-4094> svlan <2-4094> [no] bridge <1-32> address MAC forward IFNAME [no] bridge <1-32> address MAC forward IFNAME vlan <2-4094> [no] bridge <1-32> address MAC forward IFNAME vlan <2-4094> svlan <2-4094>
Parameters	**<1-32>**: Bridge group ID **MAC:** Media Access Control (MAC) address in HHHH.HHHH.HHHH format **forward**: forward matching frames **discard**: discard matching frames **IFNAME:** Interface on which the frame are sent **vlan**: VLAN ID in range of <2-4094> **svlan**: SVLAN ID in range of <2-4094>
Example usage	`switch_a(config)# bridge 2 address 2222.2222.2222 forward eth0`

bridge forward-time	
Purpose	Set the time (in seconds) after which (if this bridge is the root bridge) each port changes states to learning and forwarding. This value is used by all instances.
Command Mode	Global Configuration
Syntax	[no] bridge <1-32> forward-time <4-30>
Parameters	**<1-32>:** Bridge group ID **<4-30>**: forwarding time delay in seconds.
Example usage	`switch_a(config)# bridge 3 forward-time 6`

bridge mac-priority-override	
Purpose	Set a MAC priority override
Command Mode	Global Configuration
Syntax	[no] bridge <1-32> mac-priority-override mac-address MAC interface IFNAME vlan VLANID (static | static-priority-override | static-mgmt | static-mgmt-priority-overide) priority <0-7>
Parameters	**<1-32>**: Bridge group ID. **mac-address**: MAC address in HHHH.HHHH.HHHH format. **interface**: Interface information **IFNAME**: interface name **vlan**: add a single VLAN ID **static**: MAC is a static entry **static-mgmt**: MAC is a Static Management **static-mgmt-priority-override**: MAC is a Static Management with priority override **static-priority-override**: MAC is a static with priority override **priority**: <0-7> priority value
Example usage	`switch_a(config)# bridge 1 mac-priority-override mac-address 1111.1111.1111 interface ge1 vlan 2 static priority 80`

bridge shutdown	
Purpose	Disable a bridge
Command Mode	Global Configuration
Syntax	[no] bridge shutdown <1-32> [bridge-forward]
Parameters	**<1-32>**: the bridge group ID **bridge-forward**: put all ports of the bridge into forwarding state
Example usage	`switch_a(config)# bridge shutdown 4`
Note	Use the no parameter to reset the bridge.

bridge transmit-holdcount	
Purpose	Set the maximum number of transmissions of BPDUs by the transmit state machine. Use the no parameter to restore the default transmit hold-count value.
Command Mode	Global Configuration
Syntax	[no] bridge <1-32> transmit-holdcount <1-10>
Parameters	**<1-32>**: Bridge group ID **<1-10>**: transmit hold-count value.
Example usage	`switch_a(config)# bridge 1 transmit-holdcount 5`

bridge group	
Purpose	Bind an interface with a bridge specified by the parameter.
Command Mode	Interface Configuration
Syntax	[no] bridge-group <1-32>
Parameters	**<1-32>**: Bridge group ID
Example usage	`switch_a(config-if)# bridge-group 2`

bridge-group path-cost	
Purpose	Set the cost of a path associated with a bridge group. The lower the path cost, the greater the likelihood of the bridge becoming root. Use the no parameter to restore the default priority value.
Command Mode	Interface Configuration
Syntax	[no] bridge-group <1-32> path-cost <1-200000000>
Parameters	**<1-32>**: Bridge group ID path-cost: the path-cost of a port <1-200000000>
Example usage	`switch_a(config-if)# bridge-group 3 path-cost 123`

bridge-group priority	
Purpose	Set the port priority for a bridge. A lower priority indicates a greater likelihood of the bridge becoming root.
Command Mode	Interface Configuration
Syntax	[no] bridge-group <1-32> path-cost <1-200000000>
Parameters	<1-32> the bridge group ID path-cost: the path-cost of a port <1-200000000>
Example usage	`switch_a(config-if)# bridge-group 4 priority 96`
Note	Default priority is 1

Storm Control

storm-control	
Purpose	Use this command to set the rising threshold level for broadcast, multicast, or destination lookup failure traffic. The storm control action occurs when traffic utilization reaches this level. Storm control is used to block the forwarding of unnecessary flooded traffic. A packet storm occurs when a large number of broadcast packets are received on a port. Forwarding these packets can cause the network to slow down or time out.
Command Mode	Interface Configuration
Syntax	[no] storm-control (broadcast | multicast | dlf) level LEVEL
Parameters	**broadcast**: broadcast rate limiting **multicast**: multicast rate limiting **dlf**: destination lookup failure limiting **level**: The percentage of the threshold **LEVEL**: percentage of the maximum speed (pps) of the interface <0.00-100.00>
Example usage	`switch_a(config-if)# storm-control broadcast level 30`
Note	By default, storm control is disabled.

Storm Detect

storm-detect	
Purpose	Configured a switch to disable a port that is receiving a large number of Broadcast and/or Multicast packets. The switch can monitor for packets and take action based on percentage of bandwidth utilization or number of packets per second.
Command Mode	Global Configuration
Syntax	[no] bridge <1-32> storm-detect errdisable bridge <1-32> storm-detect interval <2-65535> bridge 1 storm-detect errdisable-recovery <0-65535>
Parameters	**<1-32>**: Bridge group ID
Example usage	`switch_a(config)# bridge 1 storm-detect errdisable`

storm-detect interval	
Purpose	Set the storm detect interval
Command Mode	Global Configuration
Syntax	bridge <1-32> storm-detect interval <2-65535>
Parameters	**<1-32>** the bridge group ID **Storm Detect interval**: <2- 65535> in seconds
Example usage	`switch_a(config)# bridge 1 storm-detect errdisable-recovery 60`
Note	Default interval is 0 (disabled).

storm-detect recovery	
Purpose	Set the Storm-Detect errdisable-recovery time. This value determines if the switch should re-enable the port after the specified value or leave the port disabled.
Command Mode	Global Configuration
Syntax	bridge 1 storm-detect errdisable-recovery <0-65535>
Parameters	**<1-32>**: Bridge group ID
Example usage	`switch_a(config)# bridge 1 storm-detect errdisable 60`

storm-detect packet type	
Purpose	Enable this port's storm detect by detect number of broadcast or broadcast plus multicast packets per second. Unit is packets per second.
Command Mode	Interface Configuration
Syntax	storm-detect (bc | mc-bc) pps <0-100000>
Parameters	**bc**: broadcast only **mc-bc**: count broadcast & multicast packets together. **pps <0-100000>**: packets per second
Example usage	`switch_a(config-if)# storm-detect mc-bc pps 50000`
Note	The Default is 0 (disabled).

storm detect utilization	
Purpose	Set the By Utilization(%) for a port. Setting this will cause the port to be disabled when the defined percentage of bandwidth is reached.
Command Mode	Interface Configuration
Syntax	storm-detect utilization <0-100>
Parameters	**<0-100>**: percentage of bandwidth
Example usage	`switch_a(config-if)# storm-detect utilization 80`
Note	The Default is 0 (disabled).

no storm-detect port enable	
Purpose	Disable storm detect on a port.
Command Mode	Interface Configuration
Syntax	no storm-detect port enable
Parameters	None
Example usage	`switch_a(config-if# no storm-detect port enable`

Trunking

show etherchannel	
Purpose	Display information about LACP channels.
Command Mode	Privileged Exec
Syntax	show etherchannel <1-65535> show etherchannel all show etherchannel detail show etherchannel load-balance show etherchannel summary
Parameters	**<1-65535>**: channel-group number.
Example usage	`switch_a# show etherchannel 5`

static-channel-group	
Purpose	Create a static aggregator, or add a member port to an existing static aggregator. This command adds the interface to the static aggregator with the specified key. If the aggregator does not exist, it is created, and the interface added to it. If the port is the last member to be detached, the static aggregator is deleted.
Command Mode	Interface Configuration
Syntax	static-channel-group <1-12> no static-channel-group
Parameters	**<1-12>**: Channel group number.
Example usage	`switch_a(config-if)# static-channel-group 2`

LACP Trunking

channel-group mode	
Purpose	Add a port to a channel group specified by the channel group number (<1-12>). This command enables link aggregation on a port, so that it may be selected for aggregation by the local system.
Command Mode	Interface Configuration
Syntax	channel-group mode <active | passive> no channel-group

Parameters	**<1-65535>**: Channel group number. **mode**: Channel mode. **active**: enable initiation of LACP negotiation on a port. **passive**: disable initiation of LACP negotiation on a port.
Example usage	`switch_a(config-if)# channel-group 4 mode active`

show lacp-counter	
Purpose	Display the packet traffic on all ports of all present LACP aggregators, or a given LACP aggregator.
Command Mode	Exec and Privileged Exec
Syntax	show lacp-counter <1-65535>
Parameters	**<1-65535>**: channel-group number.
Example usage	`switch_a# show lacp-counter 555`

show lacp sys-id	
Purpose	Display LACP system id and priority
Command Mode	Exec and Privileged Exec
Syntax	show lacp sys-id
Parameters	None
Example usage	`switch_a# show lacp sys-id`

clear lacp	
Purpose	Clear all counters of all present LACP aggregators or a given LACP aggregator.
Command Mode	Exec and Privileged Exec
Syntax	clear lacp <1-65535> counters clear lacp counters
Parameters	**<1-65535>**: channel-group number.
Example usage	`switch_a# clear lacp counters`

debug LACP	
Purpose	Turn on/off LACP debugging at various levels.
Command Mode	Exec and Privileged Exec
Syntax	[no] debug lacp (event | cli | timer | packet | sync | ha | all) [no] debug lacp timer detail
Parameters	**all**: enable all LACP debugging. **cli**: echo commands to console. **event**: set the debug options for LACP events. **ha**: echo High Availability events to console. **packet**: set the debug option for LACP packets. **sync**: echo synchronization to console. **timer**: echo timer expiry to console. **detail**: echo timer start/stop to console.
Example usage	`switch_a# debug lacp all`

lacp port-priority	
Purpose	Set the priority of a channel. Channels are selected for aggregation based on their priority with the higher priority (numerically lower) channels selected first.
Command Mode	Interface Configuration
Syntax	lacp port-priority <1-65535> no lacp port-priority
Parameters	**<1-65535>**: LACP port priority
Example usage	`switch_a(config-if)# lacp port-priority 34`

lacp system-priority	
Purpose	Set the system priority of a local system. This determines the system responsible for resolving conflicts in choice of aggregation groups.
Command Mode	Global Configuration
Syntax	lacp system-priority <1-65535> no lacp port-priority
Parameters	**<1-65535>**: LACP system priority
Example usage	`switch_a(config)# lacp system-priority 6700`
Note	Lower numerical values have higher priorities. Default system priority is 32768.

lacp timeout	
Purpose	Set either a short or long timeout value on a port. The timeout value is the number of seconds before invalidating a received LACP data unit.
Command Mode	Interface Configuration
Syntax	lacp timeout (short | long)
Parameters	**short**: LACP short timeout. Short timeout value is 3 seconds. **long**: LACP long timeout. Long timeout value is 90 seconds.
Example usage	`switch_a(config-if)# lacp timeout short`

port-channel load-balance	
Purpose	Configure LACP port-channel load-balancing and set port selection criteria (PSC) on an interface.
Command Mode	Interface Configuration
Syntax	port-channel load-balance (dst-mac | src-mac | src-dst-mac | dst-ip | src-ip | src-dstip | dst-port |src-port | src-dst-port) no port-channel load-balance
Parameters	**dst-ip**: Destination IP address-based load balancing. **dst-mac**: Destination MAC address-based load balancing. **dst-port**: Destination TCP/UDP address-based load balancing. **src-dst-ip**: Source and Destination IP address-based load balancing. **src-dst-mac**: Source and Destination MAC address-based load balancing. **src-dst-port**: Source and Destination TCP/UDP address-based load balancing. **src-ip**: Source IP address-based load balancing. **src-mac**: Source MAC address-based load balancing. **src-port**: Source port address-based load balancing.
Example usage	`switch_a(config-if)# port-channel load-balance src-dst-mac`

GVRP

show gvrp

Purpose	Display GVRP configuration, finite state machine, statistics, and timer.
Command Mode	Privileged Exec
Syntax	show gvrp configuration show gvrp machine show gvrp statistics show gvrp timer
Parameters	None
Example usage	`switch_a# show gvrp configuration`

clear gvrp	
Purpose	Clear GVRP statistics for all VLANS in a bridge or interface,
Command Mode	Privileged Exec
Syntax	clear gvrp statistics clear gvrp statistics all clear gvrp statistics bridge BRIDGE_NAME clear gvrp statistics IFNAME
Parameters	**all**: Clears a port name. **BRIDGE_NAME**: Bridge identifier. **IFNAME**: Interface name.
Example usage	`switch_a# clear gvrp statistics all`

set gvrp enable/disable	
Purpose	Enable/disable GVRP globally for a default bridge instance, not for all ports of the bridge. After enabling GVRP globally, use **set port gvrp** to enable GVRP on individual ports of the bridge.
Command Mode	Global Configuration
Syntax	set gvrp [enable | disable] set gvrp enable bridge BRIDGE_NAME set gvrp disable bridge BRIDGE_NAME
Parameters	**BRIDGE_NAME**: Bridge identifier.
Example usage	`switch_a(config)# set gvrp disable bridge 12`

set gvrp dynamic-vlan-creation	
Purpose	Enable/disable dynamic VLAN creation for default bridge instance.
Command Mode	Global Configuration
Syntax	set gvrp dynamic-vlan-creation enable set gvrp dynamic-vlan-creation enable bridge BRIDGE_NAME set gvrp dynamic-vlan-creation disable set gvrp dynamic-vlan-creation disable bridge BRIDGE_NAME
Parameters	**BRIDGE_NAME**: Bridge identifier.
Example usage	`switch_a(config)# set gvrp dynamic-vlan-creation enable bridge 2`

set gvrp registration	
Purpose	Set GVRP registration type to fixed, forbidden, or normal.
Command Mode	Global Configuration
Syntax	set gvrp registration <fixed | forbidden | normal> IF_NAME
Parameters	**fixed**: Determine that registered multicast groups are applied to the port, but that subsequent registrations or de-registrations do not affect the port. This means that none of the registered multicast groups on the port are to be de-registered based on GARP timers. **forbidden**: All GVRP multicasts are de-registered, and prevents further GVRP multicast registration on the port. **normal**: Sets dynamic GVRP multicast registration and de-registration on the port. **IF_NAME**: Name of the interface. 1 to 16 characters in length.
Example usage	`switch_a(config)# set gvrp registration fixed eth0`

set gvrp applicant	
Purpose	Set the GVRP applicant state to normal or active.
Command Mode	Global Configuration
Syntax	set gvrp applicant state normal IF_NAME set gvrp applicant state active IF_NAME
Parameters	**active**: Sets the active state. **normal**: Sets the normal state. **IF_NAME**: Name of the interface.
Example usage	`switch_a(config)# set gvrp applicant state active eth0`

set gvrp timer	
Purpose	Set the GVRP timers.
Command Mode	Global Configuration
Syntax	set gvrp timer join TIMER_VALUE IF_NAME set gvrp timer leave TIMER_VALUE IF_NAME set gvrp timer leaveall TIMER_VALUE IF_NAME
Parameters	**join**: the timer for joining the group. **leave**: the timer for leaving a group. **leaveall**: the timer for leaving all groups. **TIMER_VALUE**: timer value in hundredths of a second. **IF_NAME**: name of the interface.
Example usage	`switch_a(config)# set gvrp timer leave 245 eth0`

set port gvrp	
Purpose	Enable or disable GVRP on a port or all ports in a bridge.
Command Mode	Global Configuration
Syntax	set port gvrp enable (IF_NAME | all) set port gvrp disable (IF_NAME | all)
Parameters	**enable**: Enables GVRP on a port. **disable**: Disables GVRP on a port. **all**: All ports added to recently configured bridge. **IF_NAME**: name of the interface.
Example usage	`switch_a(config)# set port gvrp enable eth0`

7 IGMP

IGMP Information

show ip igmp	
Purpose	Show multicast groups with receivers connected and learned through IGMP. Show state of IGMP, IGMP Proxy service for specified interface, or all interfaces. Show state of IGMP Proxy services for specified interface or for all interfaces. Show Source-Specific-Multicast Mapping, VPN Routing/Forwarding instance.
Command Mode	Privileged exec
Syntax	show ip igmp groups show ip igmp (vrf NAME |) interface (IFNAME |) show ip igmp (vrf NAME |)proxy show ip igmp ssm-map show ip igmp vrf
Parameters	**NAME**: VPN Routing/Forwarding instance name
Example usage	`switch_a# show ip igmp vrf`

clear ip igmp	
Purpose	Clear IGMP local-memberships on interfaces.
Command Mode	Privileged exec
Syntax	clear ip igmp clear ip igmp group * clear ip igmp group A.B.C.D clear ip igmp group A.B.C.D IFNAME clear ip igmp interface IFNAME clear ip igmp (vrf NAME |) clear ip igmp (vrf NAME |) group * clear ip igmp (vrf NAME |) group A.B.C.D clear ip igmp (vrf NAME |) group A.B.C.D IFNAME clear ip igmp (vrf NAME |) interface IFNAME
Parameters	***:** Clears all groups on all interfaces. **A.B.C.D**: Group address's local-membership cleared from all interfaces. **interface**: All groups learned from this interface are deleted. **vrf**: The VRF name. **group**: Deletes IGMP group cache entries.
Example usage	`switch_a# clear ip igmp interface eth1`

ip multicast-routing	
Purpose	Turn on/off multicast routing on the router
Command Mode	Global configuration
Syntax	[no] ip multicast-routing [no] ip multicast-routing (vrf NAME |)
Parameters	**NAME**: VPN Routing/Forwarding instance name
Example usage	`switch_a(config)# no ip multicast-routing`

ip igmp	
Purpose	Enable the IGMP protocol operation on an interface. This command enables IGMP protocol operation in stand-alone mode, and can be used to learn local-membership information prior to enabling a multicast routing protocol on the interface. This command will has no effect on interfaces configured for IGMP Proxy. Use the no parameter to return all IGMP related configuration to the default (including IGMP Snooping or IGMP Proxy service).
Command Mode	Interface Configuration
Syntax	[no] ip igmp
Parameters	none
Example usage	`switch_a(config-if)# ip igmp`

ip igmp version	
Purpose	Set the current IGMP protocol version on an interface. This command applies to interfaces configured for IGMP Layer-3 multicast protocols, IGMP Snooping, or IGMP Proxy. Use the no parameter to return to the default version.
Command Mode	Interface Configuration
Syntax	ip igmp version <1-3> no ip igmp version
Parameters	<1-3>: IGMP version number.
Example usage	`switch_a(config-if)# ip igmp version 2`

ip igmp join-group	
Purpose	Configure a join multicast group. Use the no parameter to delete group membership entry.
Command Mode	Global Configuration
Syntax	ip igmp join-group A.B.C.D {(source (A.B.C.D)|)} no ip igmp join-group A.B.C.D {(source (A.B.C.D)|)}
Parameters	**A.B.C.D**: Standard IP multicast group address to be configured as a group member. **source**: Static source to be joined. **A.B.C.D**: Standard IP source address to be configured as a source from where multicast packets originate.
Example usage	`switch_a(config-if)# ip igmp join-group 1.1.1.1 source 1.1.1.2`

ip igmp proxy-service	
Purpose	Designate an interface to be the IGMP proxy-service (upstream host-side) interface, thus enabling IGMP host-side protocol operation on this interface. All associated downstream router-side interfaces will have their memberships consolidated on this interface, according to IGMP host-side functionality.
Command Mode	Interface Configuration
Syntax	[no] ip igmp proxy-service
Parameters	None
Example usage	`switch_a(config-if)# ip igmp join-group 1.1.1.1 source 1.1.1.2`
Note	This command should not be used when configuring interfaces enabled for IGMP in association with a multicast-routing protocol, otherwise the behavior will be undefined.

ip igmp mroute-proxy	
Purpose	Specify the IGMP Proxy service (upstream host-side) interface with which to be associated. IGMP router-side protocol operation is enabled only when the specified upstream proxy-service interface is functional.
Command Mode	Interface Configuration
Syntax	ip igmp mroute-proxy IFNAME no ip igmp mroute-proxy
Parameters	**IFNAME**: Interface name
Example usage	`switch_a(config-if)# ip igmp mroute-proxy ge5`
Note	This command should not be used when configuring interfaces enabled for IGMP in association with a multicast-routing protocol, otherwise the behavior will be undefined.

ip igmp immediate-leave	
Purpose	In IGMP version 2, use this command to minimize the leave latency of IGMP memberships. This command is used when only one receiver host is connected to each interface. This command applies to interfaces configured for IGMP Layer-3 multicast protocols, IGMP Snooping, or IGMP Proxy.
Command Mode	Interface Configuration
Syntax	ip igmp immediate-leave group-list (<1-99> | <1300-1999> | WORD) no ip igmp immediate-leave
Parameters	**group-list**: Standard access-list name or number that defines multicast groups in which the immediate leave feature is enabled. **<1-99>**: Access-list number. **<1300-1999>**: Access-list number (expanded range). **WORD**: Standard IP access-list name.
Example usage	`switch_a(config-if)# ip igmp immediate-leave group-list 34`

ip igmp access-group	
Purpose	Control the multicast local-membership groups learnt on an interface. This command applies to interfaces configured for IGMP Layer-3 multicast protocols, IGMP Snooping, or IGMP Proxy.
Command Mode	Interface Configuration
Syntax	ip igmp access-group (<1-99> | WORD) no ip igmp access-group
Parameters	**<1-99>**: Access-list number. **WORD**: Standard IP access-list name.

Example usage	In this example, hosts serviced by interface ge5 can only join the group 225.2.2.2:
	`switch_a#configure terminal`
	`switch_a(config)#access-list 1 permit 225.2.2.2 0.0.0.0`
	`switch_a(config)#interface ge5`
	`switch_a(config-if)#ip igmp access-group 1`

ip igmp limit	
Purpose	Configure limit for maximum number of group membership states, at router level, or for the specified interface. Once the specified number of group memberships is reached, additional local memberships are ignored. An exception access-list can be used to specify group-addresses to be excluded from the limit. This command is for interfaces configured for IGMP Layer-3 multicast protocols, IGMP Snooping, or IGMP Proxy. When configured for IGMP Snooping, this command can be issued on only VLAN interfaces. The limit applies, individually, to each of its constituent interfaces.
Command Mode	Global configuration and Interface Configuration
Syntax	ip igmp limit (<1-2097152> (except (<1-99>|<1300-1999>| WORD)|) ip igmp limit (vrf NAME)(<1-2097152> (except (<1-99> | <1300-1999> | WORD) |) no ip igmp limit no ip igmp (vrf NAME |) limit
Parameters	**vrf**: VRF name. **<1-2097152>**: Max number of group membership states. **except**: Multicast groups that are exempted from limit. **<1-99>**: Access-list number, **<1300-1999>**: Access-list number (expanded range), **WORD**: Standard IP access-list name.

Example usage	Set IGMP limit of 100 group-membership states across all interfaces on which IGMP is enabled, and excludes group 224.1.1.1 from this limitation.
	`switch_a(config)# access-list 1 permit 224.1.1.1 0.0.0.0`
	`switch_a(config)# ip igmp limit 100 except 1`

IGMP Snooping

show igmp snooping	
Purpose	Show IGMP multicast group membership, interface information, multicast router information, and statistics.
Command Mode	Privileged exec
Syntax	show igmp snooping group show igmp snooping interface show igmp snooping mrouter show igmp snooping statistics
Parameters	None
Example usage	`switch_a# show igmp snooping statistics`

ip igmp snooping enable	
Purpose	Enable IGMP Snooping. (non-querier role)
Command Mode	Global configuration
Syntax	ip igmp snooping enable no ip igmp snooping
Parameters	None
Example usage	`switch_a(config)# ip igmp snooping enable`

ip igmp snooping querier	
Purpose	Enable IGMP snooping querier role.
Command Mode	Global configuration
Syntax	ip igmp snooping querier no ip igmp snooping querier
Parameters	None
Example usage	`switch_a(config)# igmp snooping querier`

igmp snooping report-suppression	
Purpose	Enable Report suppression on global level.
Command Mode	Global configuration
Syntax	igmp snooping report-suppression no igmp snooping report-suppression
Parameters	None
Example usage	`switch_a(config)# igmp snooping report-suppression`

ip igmp snooping force-forward	
Purpose	Control how the switch will forward unknown multicast packets when the switch is in IGMP Passive mode, both with or without a Querier Port present.
Command Mode	Global configuration
Syntax	ip igmp snooping force-forward (LINE | all | none)
Parameters	**LINE**: Do not forward multicast packets to any interface **all**: Flood all unknown multicast packets **none**: Forward all multicast packets to all interfaces
Example usage	`switch_a(config-)# igmp snooping force-forward all`

ip igmp snooping passive-forward	
Purpose	Control how the switch will forward unknown multicast packets when the switch is in IGMP Passive mode and also without a Querier Port present.
Command Mode	Global configuration
Syntax	ip igmp snooping passive-forward (LINE | all | none)
Parameters	**LINE**: Do not forward multicast packets to any interface **all**: Flood all unknown multicast packets **none**: Drop all unknown multicast packets
Example usage	`switch_a(config-if)# igmp snooping passive-forward none`

ip igmp snooping	
Purpose	Enable IGMP Snooping in the VLAN interface.
Command Mode	Interface mode for VLAN
Syntax	[no] ip igmp snooping
Parameters	None
Example usage	`switch_a(config-if)# ip igmp snooping`

igmp snooping fast-leave	
Purpose	Enable IGMP Snooping fast-leave processing. Fast-leave processing is analogous to immediate leave processing; the IGMP group-membership is removed as soon as an IGMP leave group message is received without sending out a group-specific query.
Command Mode	Interface mode for VLAN
Syntax	[no] igmp snooping fast-leave
Parameters	None
Example usage	`switch_a(config-if)# ip igmp snooping fast-leave`

igmp snooping mrouter	
Purpose	Configure the specified VLAN constituent interface as a multicast router interface for IGMP Snooping in that VLAN.
Command Mode	Interface mode for VLAN
Syntax	[no] igmp snooping mrouter interface IFNAME
Parameters	**IFNAME**: Name of the interface
Example usage	`switch_a(config-if)# igmp snooping mrouter interface ge8`

ip igmp snooping querier	
Purpose	Enable IGMP snooping querier functionality in VLAN interface.
Command Mode	Interface mode for VLAN
Syntax	ip igmp snooping querier no ip igmp snooping querier
Parameters	None
Example usage	`switch_a(config-if)# ip igmp snooping querier`

igmp snooping report-suppression	
Purpose	Enable report suppression for IGMP version 1, 2 and 3 reports.
Command Mode	Interface mode for VLAN
Syntax	igmp snooping report-suppression no igmp snooping report-suppression
Parameters	None
Example usage	`switch_a(config-if)# igmp snooping report-suppression`

igmp snooping static-group	
Purpose	Configure an interface belonging to a VLAN as a static member of a multicast group. Interface can be specified by type and number.
Command Mode	Interface mode for VLAN
Syntax	[no] igmp snooping static-group A.B.C.D [source A.B.C.D] interface IFNAME
Parameters	None
Example usage	`switch_a(config-if)# igmp snooping static-group 230.0.0.1 interface ge10`

GMRP

show gmrp	
Purpose	Show GMRP configurations, GMRP finite state machine, GMRP statistics, and GMRP timer.
Command Mode	Privileged exec
Syntax	show gmrp configuration show gmrp machine show gmrp statistics show gmrp timer
Parameters	None
Example usage	`switch_a# show gmrp configuration`

clear gmrp statistics	
Purpose	Clear GMRP statistics for a given VLAN or all the VLANs .
Command Mode	Privileged exec
Syntax	clear gmrp statistics all clear gmrp statistics all bridge BRIDGE_NAME clear gmrp statistics vlanid <1-4094> clear gmrp statistics vlanid <1-4094> bridge <1-32>
Parameters	**BRIDGE_NAME**: Bridge identifier. **<1-4094>**: VLAN identifiers, **<1-32>**: bridge identifier.
Example usage	`switch_a# clear gmrp statistics vlan 12 bridge 2`

set gmrp enable	
Purpose	Enable GMRP globally on a switch for the default bridge. This command does not enable GMRP for all ports of the bridge. After enabling GMRP globally, use the **set port gmrp** command to enable GMRP on individual ports. GMRP cannot be enabled if IGMP Snooping is enabled or if GMRP is configured for a VLAN.
Command Mode	Global Configuration
Syntax	set gmrp enable set gmrp enable bridge BRIDGE_NAME set gmrp enable bridge BRIDGE_NAME vlan VLANID set gmrp enable vlan VLANID
Parameters	**BRIDGE_NAME**: Bridge identifier. **VLANID**: VLAN identifier <1-4094>.
Example usage	`switch_a(config)# set gmrp enable bridge 2`

set gmrp disable	
Purpose	Disable GMRP globally on a switch for the default bridge. This command does not disable GMRP in all ports of the bridge.
Command Mode	Global Configuration
Syntax	set gmrp disable set gmrp disable bridge BRIDGE_NAME set gmrp disable bridge BRIDGE_NAME vlan VLANID set gmrp disable vlan VLANID
Parameters	**BRIDGE_NAME**: Bridge identifier. **VLANID**: VLAN identifier <1-4094>.
Example usage	`switch_a(config)# set gmrp disable bridge 2 vlan 2`

set gmrp extended-filtering	
Purpose	Enable or disable extended filtering on a bridge as per Table 8-7 of IEEE802.1Q-2003.
Command Mode	Global Configuration
Syntax	set gmrp extended-filtering enable set gmrp extended-filtering enable bridge BRIDGE_NAME set gmrp extended-filtering disable set gmrp extended-filtering disable bridge BRIDGE_NAME
Parameters	enable: Enables GMRP on a switch. disable: Disables GMRP on a switch. BRIDGE_NAME: Bridge identifier.
Example usage	`switch_a(config)#set gmrp extended-filtering enable`

set gmrp fwdall	
Purpose	Set the **GMRP forward all** option for an interface.
Command Mode	Global Configuration
Syntax	set gmrp fwdall disable IF_NAME set gmrp fwdall enable IF_NAME
Parameters	enable: Enables GMRP on a switch. disable: Disables GMRP on a switch. IF_NAME: Interface name.
Example usage	`switch_a(config)# set gmrp fwdall enable ge5`

set gmrp registration	
Purpose	Set GMRP registration type. To de-register a multicast port, the port must be in the **normal** registration mode.
Command Mode	Global Configuration
Syntax	set gmrp registration fixed IF_NAME set gmrp registration forbidden IF_NAME set gmrp registration normal IF_NAME set gmrp registration restricted IF_NAME
Parameters	**fixed**: Determine that the multicast groups currently registered on the switch are applied to the port, but that subsequent registrations or de-registrations do not affect the port. This means that none of the registered multicast groups on the port are to be de-registered based on GARP timers. **forbidden**: All GMRP multicasts are de-registered, and prevents further GMRP multicast registration on the port. **normal**: dynamic GMRP multicast registration and de-registration on the port. **restricted**: Restricted registration. **IF_NAME**: Interface name, from 1 to 16 characters in length
Example usage	`switch_a(config)# set gmrp registration normal ge5 bridge 2`

set gmrp timer	
Purpose	Set the values for the GMRP Join, Leave, and Leaveall timers for a specified bridge. The relationship for the timer values are as follows: • Leave timer must be greater than, or equal to, three times the join timer. • Leaveall timer must be greater than the leave timer.
Command Mode	Global Configuration
Syntax	set gmrp timer join TIMER_VALUE IF_NAME set gmrp timer leave TIMER_VALUE IF_NAME set gmrp timer leaveall TIMER_VALUE IF_NAME
Parameters	**join**: The timer for joining the group. **leave**: The timer for leaving a group. **leaveall**: The timer for leaving all groups. **TIMER_VALUE**: Timer value in hundredths of a second. **IF_NAME**: Interface name.
Example usage	`switch_a(config)# set gmrp timer join 100 eth0`
Note	Default for the join timer is 200 milliseconds (ms). Default for the leave timer is 600 milliseconds. Default for the leaveall timer is 10000 ms.

set port gmrp	
Purpose	Enable or disable GMRP on a particular port in all VLANs or all ports in a bridge. GMRP on a port cannot be enabled for all VLANs if GMRP has already been configured for a particular VLAN for the port.
Command Mode	Global Configuration
Syntax	set port gmrp disable (IF_NAME|all) set port gmrp enable (IF_NAME|all) set port gmrp disable IF_NAME vlan VLANID set port gmrp enable IF_NAME vlan VLANID
Parameters	**enable**: Enables GMRP on a switch. **disable**: Disables GMRP on a switch. **all**: All ports added to recently configured bridge. **IFNAME**: Interface name **VLANID**: VLAN identifier <1-4094>.
Example usage	`switch_a(config)# set port gmrp enable eth0`

8 STP

STP Information

show spanning-tree	
Purpose	Show the state of the spanning tree for all STP or RSTP bridge-groups, including named interface and VLANs.
Command Mode	Privileged exec
Syntax	show spanning-tree show spanning-tree interface IFNAME show spanning-tree mst show spanning-tree mst config show spanning-tree mst interface IFNAME show spanning-tree mst detail show spanning-tree mst detail interface IFNAME show spanning-tree mst instance (<1-63> | spbm) interface IFNAME show spanning-tree mst instance (<1-63> | spbm | te-msti) show spanning-tree rpvst+ show spanning-tree rpvst+ config show spanning-tree rpvst+ detail show spanning-tree rpvst+ detail interface IFNAME show spanning-tree rpvst+ interface IFNAME show spanning-tree rpvst+ vlan <1-4094> show spanning-tree rpvst+ vlan <1-4094> interface IFNAME show spanning-tree statistics bridge <1-32> show spanning-tree statistics interface IFNAME (instance (<1-63> | spbm) | vlan <2-4094>) bridge <1-32> show spanning-tree statistics (interface IFNAME | (instance (<1-63> | spbm) | vlan <1-4094>)) bridge <1-32> show spanning-tree vlan range-index
Parameters	**interface**: interface information **IFNAME**: Interface name **mst**: Display MST information **rpvst+**: Display RPVST information **statistics**: Display statistics of the BPDUs **vlan range-index**: Display a VLAN range-index value
Example usage	`switch_a# show spanning-tree`

Global Configuration

bridge protocol ieee	
Purpose	Add an IEEE 802.1d Spanning Tree Protocol bridge. After creating a bridge instance, add interfaces to the bridge using the **bridge-group** command. Bring the bridge instance into operation with the **no shutdown** command in interface mode.
Command Mode	Global Configuration
Syntax	bridge <1-32> protocol ieee (vlan-bridge|) no bridge <1-32> protocol ieee
Parameters	**<1-32>**: Bridge group ID, **vlan-bridge**: Specify VLAN-aware bridge.
Example usage	`switch_a(config)# bridge <1-32> protocol ieee`

bridge spanning-tree	
Purpose	Enable/disable Spanning Tree Protocol on a bridge.
Command Mode	Global Configuration
Syntax	bridge <1-32> spanning-tree enable no bridge <1-32> spanning-tree enable (bridge-forward|)
Parameters	**<1-32>**: Bridge group ID. **enable**: Enable spanning tree protocol on this bridge. **bridge-forward**: Puts all ports of bridge into forwarding state.
Example usage	`switch_a(config)# bridge 2 spanning-tree enable`

bridge spanning-tree errdisable-timeout	
Purpose	Enable the error-disable-timeout facility, which sets a timeout for ports disabled by the BPDU guard feature.The timer sets the interval ro the port to be anabled back.
Command Mode	Global Configuration
Syntax	bridge <1-32> spanning-tree errdisable-timeout enable bridge <1-32> spanning-tree errdisable-timeout interval <10-1000000> no bridge <1-32> spanning-tree errdisable-timeout enable no bridge <1-32> spanning-tree errdisable-timeout interval
Parameters	**enable**: Enable the timeout mechanism for the port **interval:** The interval after which port shall be enabled. **<10-1000000>**: Error-disable-timeout interval in seconds.
Example usage	`switch_a(config)# bridge 1 spanning-tree errdisable-timeout enable`

bridge spanning-tree force-version	
Purpose	Set the version for the bridge. A version identifier of less than a value of 2 enforces the spanning tree protocol. Although the command supports an input range of 0-4, for RSTP, the valid range is 0-2. When the forceversion is set for a bridge, all ports of the bridge have the same spanning tree version set. Use the **show spanning tree** command to display administratively configured and currently running values of the BPDU filter parameter for the bridge and port.
Command Mode	Global Configuration
Syntax	bridge <1-32> spanning-tree force-version <0-4> no bridge <1-32> spanning-tree force-version
Parameters	**<1-32>**: Bridge group ID. **force-version**: Specify a force version identifier: 0 STP 1 Not supported 2 RSTP 3 MSTP 4 SPB
Example usage	`switch_a(config)# bridge 1 spanning-tree force-version 0`

bridge spanning-tree pathcost	
Purpose	Set a spanning-tree path cost method. If the short parameter is used, the switch uses a value for the default path cost a number in the range 1 through 65,535. If the long parameter is used, the switch uses a value for the default path cost a number in the range 1 through 200,000,000. Use the no option to return the path cost method to the default setting. The default path cost method for STP is short and for MSTP/RSTP is long.
Command Mode	Global Configuration
Syntax	bridge <1-32> spanning-tree pathcost method (short | long) no bridge <1-32> spanning-tree pathcost method
Parameters	**<1-32>:** The bridge group ID. **method:** Method used to calculate default port path cost. **long:** 16-bit values for default port path costs. **short:** 32-bit values for default port path costs.
Example usage	`switch_a(config)#bridge 1 spanning-tree pathcost method short`

bridge spanning-tree portfast	
Purpose	Set the portfast BPDU (Bridge Protocol Data Unit) guard or filter for the bridge. Use the **show spanning tree** command to display administratively configured and currently running values of the BPDU filter parameter. **BPDU Filter** — All ports that have their BPDU filter set to default take the same value of BPDU filter as that of the bridge. The Spanning Tree Protocol sends BPDUs from all ports. Enabling the BPDU Filter feature ensures that PortFast-enabled ports do not transmit or receive any BPDUs. **BPDU Guard** — When the BPDU guard feature is set for a bridge, all portfast-enabled ports of the bridge that have the BPDU guard set to default shut down the port on receiving a BPDU. In this case, the BPDU is not processed. The port can be brought back up manually with the **no shutdown** command, or the **errdisable-timeout** feature can be used to to re-enable the port after a specified time interval.
Command Mode	Global Configuration
Syntax	[no] bridge <1-32> spanning-tree portfast bpdu-guard [no] bridge <1-32> spanning-tree portfast bpdu-filter
Parameters	**<1-32>**: Bridge group ID. **bpdu-filter**: Filter the BPDUs on portfast enabled ports. **bpdu-guard**: Guard the portfast ports against BPDU receive.
Example usage	`switch_a(config)# bridge 3 spanning-tree portfast bpdu-filter`

bridge vlan priority	
Purpose	Create or delete a mapping between an MSTI and VLAN for RPVST+ operation. The bridge instance must already be configured for RPVST+ operation. This command sets the priority value for the spanning-tree on the bridge. The lower the priority of the VLAN on a bridge, the better the chances of the bridge becoming a root bridge, or a designated bridge for the VLAN. The permitted range of values is 0-61440. The no command resets to default **priority** (32768).
Command Mode	Global Configuration
Syntax	bridge <1-32> vlan <2-4094> priority <0-61440> no bridge <1-32> vlan <2-4094> priority
Parameters	**<1-32>**: Bridge group ID, **vlan**: Identity of VLAN <2-4094>. **priority**: The bridge priority for the common instance. **<0-61440>**: Bridge priority in increments of 4096

Example usage	`switch_a(config)# bridge 1 vlan 2 priority 80`
bridge hello-time	
Purpose	Set the hello-time, the time in seconds after which (if this bridge is the root bridge) all the bridges in a bridged LAN exchange Bridge Protocol Data Units (BPDUs). A very low value of this parameter leads to excessive traffic on the network, while a higher value delays the detection of topology change.This value is used by all instances. Configure the bridge instance name before using this command. The allowable range of values is 1-10 seconds. Make sure that the value of hello time is always greater than the value of hold time (2 seconds by default). Use the no parameter to restore the default value of the hello time. Default hello time value is 2 seconds.
Command Mode	Global Configuration
Syntax	bridge <1-32> hello-time <1-10> no bridge <1-32> hello-time
Parameters	**<1-32>**: Bridge group ID. **<1-10>**: Hello BPDU interval in seconds.
Example usage	`switch_a(config)# bridge 3 hello-time 3`

bridge priority	
Purpose	Set the bridge priority for the common instance. Using a lower priority indicates a greater likelihood of the bridge becoming root. The priority values can be set only in increments of 4096. Use the no parameter to reset it to the default value. The default priority is 32768 (or hex 0x8000).
Command Mode	Global Configuration
Syntax	bridge (<1-32> |) priority <0-61440> no bridge (<1-32> |)
Parameters	**<1-32>**: Bridge group ID. **<0-61440>**: Bridge priority
Example usage	`switch_a(config)# bridge 2 priority 4096`

bridge max-age	
Purpose	Set the maximum age for a bridge. This value is used by all instances. Maximum age is the maximum time in seconds for which (if a bridge is the root bridge) a message is considered valid. This prevents the frames from looping indefinitely. The value of maximum age should be greater than twice the value of hello time plus 1, but less than twice the value of forward delay minus 1. The allowable range for max-age is 6-40 seconds. Configure this value sufficiently high, so that a frame generated by root can be propagated to the leaf nodes without exceeding the maximum age. Use the no parameter to restore the default value of the maximum age (20 seconds).
Command Mode	Global Configuration
Syntax	bridge <1-32> max-age <6-40> no bridge <1-32> max-age
Parameters	**<1-32>**: Bridge group ID. **<6-40>**: Maximum time, in seconds, to listen for the root bridge.
Example usage	`switch_a(config)# bridge 2 max-age 12`

bridge forward-time	
Purpose	Set the time (in seconds) after which (if this bridge is the root bridge) each port changes states to learning and forwarding. This value is used by all instances. Use the no parameter to restore the default value (15 seconds).
Command Mode	Global Configuration
Syntax	bridge <1-32> forward-time <4-30> no bridge <1-32> forward-time
Parameters	**<1-32>**: Bridge group ID. **<6-40>**: Maximum time, in seconds, to listen for the root bridge.
Example usage	`switch_a(config)# bridge 2 max-age 12`
Note	Be careful when setting this value lower than 7 seconds

spanning-tree acquire	
Purpose	Enable the default bridge to learn station location information for an instance. This helps in making forwarding decisions.
Command Mode	Global Configuration
Syntax	[no] spanning-tree acquire
Parameters	None
Example usage	`switch_a(config)# spanning-tree acquire`
Note	Learning is enabled by default for all instances.

bridge-group spanning-tree	
Purpose	Enable or disable the spanning-tree on a configured bridge.
Command Mode	Interface Configuration
Syntax	bridge-group <1-32> spanning-tree (disable | enable)
Parameters	**<1-32>**: Bridge group ID. **disable**: Disable spanning tree on the interface. **enable**: Enable spanning tree on the interface.
Example usage	`switch_a(config-if)# bridge-group 1 spanning-tree enable`
Note	Spanning-tree is enabled by default.

bridge-group path-cost	
Purpose	Set the cost of a path. Before setting a path-cost in a VLAN configuration, add an MST instance to a port using the **bridge-group instance** command. Use the no parameter to restore the default cost value of the path, which varies according to bandwidth.
Command Mode	Interface Configuration
Syntax	bridge-group <1-32> path-cost <1-200000000> no bridge-group <1-32> path-cost
Parameters	**<1-32>**: Bridge group ID. **<1-200000000>**: Cost of the path (lower means a greater likelihood of the interface becoming root).
Example usage	`switch_a(config-if)# bridge-group 4 path-cost 1000`
Note	Assuming a 10 Mb/s link speed, the default value is 200,000.

bridge-group priority	
Purpose	Set the port priority for a bridge. A lower priority indicates a greater likelihood of the bridge becoming root.
Command Mode	Interface Configuration
Syntax	bridge-group <1-32> priority <0-240>
Parameters	**<1-32>**: Bridge group ID. **<0-240>**: Port priority, . The priority values can only be set in increments of 16.
Example usage	`switch_a(config-if)# bridge-group 4 priority 32`

bridge-group instance	
Purpose	Assign a Multiple Spanning Tree (MST) instance to a port.
Command Mode	Interface Configuration
Syntax	[no] bridge-group (<1-32> | backbone) instance (<1-63> | spbm | te-msti)
Parameters	**<1-32>**: Bridge identifier. **<1-63>**: Multiple spanning tree instance identifier. **spbm**: Shortest Path Bridging - MAC instance. **te-msti**: Traffic engineering MSTI instance.
Example usage	`switch_a(config-if)# bridge-group 1 instance te-msti`

bridge-group instance path-cost	
Purpose	Set a path cost for a multiple spanning tree instance. Before using this command, add an MST instance to a port using the **bridge-group instance** command. Use the no form to set the path cost to default, which varies according to bandwidth.
Command Mode	Interface Configuration
Syntax	bridge-group (<1-32> | backbone) instance <1-63> path-cost <1-200000000> no bridge-group (<1-32> | backbone) instance <1-63> path-cost
Parameters	**<1-32>**: Bridge identifier. **<1-63>**: Multiple spanning tree instance identifier. **<1-200000000>**: Path cost for a port (lower path cost means greater likelihood of becoming root).
Example usage	`switch_a(config-if)# bridge-group 4 instance 3 path-cost 1000`

bridge-group instance priority	
Purpose	Set the priority of a multiple spanning tree instance. The Multiple Spanning Tree Protocol uses port priority as a tiebreaker to determine which port should forward frames for a particular instance on a LAN, or which port should be the root port for an instance. A lower value implies a better priority. In the case of the same priority, the interface index will serve as the tiebreaker, with the lower-numbered interface being preferred over others.
Command Mode	Interface Configuration
Syntax	bridge-group (<1-32>) instance (<1-63>) priority <0-240>
Parameters	**<1-32>**: Bridge identifier. **<1-63>**: Multiple spanning tree instance identifier. **<0-240>**: Port priority, set in increments of 16.

Example usage	`switch_a(config-if)# bridge-group 2`
	`switch_a(config-if)# bridge-group 2 instance 4`
	`switch_a(config-if)# bridge-group 2 instance 4 priority 64`

Note	Default value of port priority is 128.

spanning tree autoedge	
Purpose	Set automatic identification of the edge port.
Command Mode	Interface Configuration
Syntax	[no] spanning-tree autoedge
Parameters	None
Example usage	`switch_a(config-if)# spanning tree autoedge`

spanning tree edgeport	
Purpose	Set a port as an edge-port and to enable rapid transitions. Use the no parameter to set port to default state (not an edge-port) and to disable rapid transitions. This command is an alias to the **spanning-tree portfast** command, can be used interchangeably.
Command Mode	Interface Configuration
Syntax	[no] spanning-tree edgeport
Parameters	None
Example usage	`switch_a(config-if)# spanning tree edgeport`

spanning tree guard root	
Purpose	Enable the root guard feature for the port. This feature disables reception of superior BPDUs. The root guard feature makes sure that the port on which it is enabled is a designated port. If the root guard enabled port receives a superior BPDU, it goes to a Listening state (for STP) or discarding state (for RSTP and MSTP).
Command Mode	Interface Configuration
Syntax	[no] spanning-tree guard root
Parameters	None
Example usage	`switch_a(config-if)# spanning tree guard root`

spanning tree portfast	
Purpose	Enable fast transitions, with port placed in the forwarding state immediately. Port does not go through listening, learning, and forwarding states.
Command Mode	Interface Configuration
Syntax	[no] spanning-tree portfast
Parameters	None
Example usage	`switch_a(config-if)# spanning tree portfast`

spanning tree bpdu-guard	
Purpose	Enable or disable the BPDU Guard feature on a port. This command supersedes the bridge level configuration for the BPDU Guard feature. When the enable or disable parameter is used with this command, this configuration takes precedence over bridge configuration. However, when the default parameter is used with this command, the bridge-level BPDU Guard configuration takes effect. Use the **show spanning tree** command to display currently running values of the BPDU filter parameter.
Command Mode	Interface Configuration
Syntax	spanning-tree bpdu-guard (enable | disable | default) no spanning-tree bpdu-guard
Parameters	None:
Example usage	`switch_a(config-if)# spanning-tree bpdu-guard enable`

spanning tree hello-time	
Purpose	Set the hello-time, the time in seconds after which (if this bridge is the root bridge) all the default bridges in a bridged LAN exchange Bridge Protocol Data Units (BPDUs). A very low value of this parameter leads to excessive traffic on the network, while a higher value delays the detection of topology change. This value is used by all instances. Use the no parameter to return to the default value for the hello time.
Command Mode	Interface Configuration
Syntax	spanning-tree hello-time <1-10> no spanning-tree hello-time
Parameters	**<1-10>**: BPDU in seconds
Example usage	`switch_a(config-if)# spanning-tree hello-time 5`
Note	Default hello time is 2. The value of the hello time must always be greater than the value of the hold time.

spanning tree enable/disable	
Purpose	Enable or disable spanning tree on an interface for the default bridge.
Command Mode	Interface Configuration
Syntax	spanning-tree enable spanning-tree disable
Parameters	None
Example usage	`switch_a(config-if)# spanning-tree enable`
Note	Spanning-tree is enabled by default if the **switchport** command is configured.

spanning-tree instance restricted-role	
Purpose	Set the restricted role value for the instance to TRUE. Use the no parameter to set the restricted role to default (FALSE).
Command Mode	Interface Configuration
Syntax	[no] spanning-tree instance <1-63> restricted-role
Parameters	**<1-63>**: Instance ID
Example usage	`switch_a(config-if)# spanning-tree instance 2 restricted-role`

spanning-tree instance restricted-tcn	
Purpose	Set the restricted Topology Change Notification (TCN) value for the instance to TRUE. Use the no parameter to set the restricted role value for the instance to default (FALSE).
Command Mode	Interface Configuration
Syntax	[no] spanning-tree instance <1-63> restricted-tcn
Parameters	**<1-63>**: Instance ID
Example usage	`switch_a(config-if)# spanning-tree instance 2 restricted-tcn`

spanning-tree link-type	
Purpose	Enable or disable point-to-point or shared link types. RSTP has a backward-compatible STP mode, spanning-tree link-type **shared**. An alternative is the spanning-tree **force-version 0**.
Command Mode	Interface Configuration
Syntax	spanning-tree link-type (auto | point-to-point | shared) no spanning-tree link-type
Parameters	**auto**: Sets to either point-to-point or shared based on duplex state. **point-to-point**: Enables rapid transition. **shared**: Disables rapid transition.
Example usage	`switch_a(config-if)# spanning-tree link-type point-to-point`

spanning-tree bpdu-filter	
Purpose	Set the BPDU filter value for individual ports. The **enable** or **disable** parameters cause this configuration to take precedence over bridge configuration. The **default** parameter causes the bridge level BPDU filter configuration to take effect for the port. Use the **show spanning tree command** to display currently running values of the BPDU filter parameter.
Command Mode	Interface Configuration
Syntax	[no] spanning-tree bpdu-filter (enable | disable | default)
Parameters	**default**: Sets the bpdu-filter to the default level. **disable**: Disables the BPDU-filter. **enable**: Enables the BPDU-filter.
Example usage	`switch_a(config-if)# spanning-tree bpdu-filter enable`

spanning-tree restricted-role	
Purpose	Set the restricted role value for the instance to TRUE. Use the no parameter to set the restricted role value for the instance to FALSE.
Command Mode	Interface Configuration
Syntax	[no] spanning-tree restricted-role
Parameters	None
Example usage	`switch_a(config-if)# spanning-tree restricted-role`
Note	Default restricted role value is FALSE.

spanning-tree restricted-tcn	
Purpose	Set the restricted TCN value for the instance to TRUE, restricting the Topology Change Notification (TCN) BPDUs sent on the port. Use the no parameter to set the restricted role value for the instance to FALSE.
Command Mode	Interface Configuration
Syntax	[no] spanning-tree restricted-tcn
Parameters	None
Example usage	`switch_a(config-if)# spanning-tree restricted-tcn`
Note	Default restricted tcn value is FALSE.

spanning-tree vlan	
Purpose	Set restrictions for the port of a particular VLAN.
Command Mode	Interface Configuration
Syntax	spanning-tree vlan <2-4094> restricted-role spanning-tree vlan <2-4094> restricted-tcn no spanning-tree vlan <2-4094> restricted-role no spanning-tree vlan <2-4094> restricted-tcn
Parameters	**<2-4094>**: VLAN identifier. **restricted-role**: Restrict the role of the port **restricted-tcn**: Restrict propagation of topology change notifications from the port
Example usage	`switch_a(config-if)# spanning-tree vlan 3 restricted-role`

traffic-class-table	
Purpose	Set the user priority and number of supported traffic classes.
Command Mode	Interface Configuration
Syntax	traffic-class-table user-priority <0-7> num-traffic-classes <1-8> value <0-7> traffic-class-table user-priority <0-7> value <0-3>
Parameters	**user-priority**: User priority associated with the traffic class table **<0-7>**: User priority value **num-traffic-classes**: Number of traffic classes **<1-8>**: Number of traffic classes **value**: Value for the given user priority/num traffic classes **<0-7>**: Value for the given user priority classes **<0-3>**: Value for the given user priority classes
Example usage	`switch_a(config-if)# traffic-class-table user-priority 3 num-traffic-classes 4 value 5`

user-priority	
Purpose	Set the default user priority associated with the interface.
Command Mode	Interface Configuration
Syntax	user-priority <0-7> no user-priority
Parameters	**<0-7>**: User priority value
Example usage	`switch_a(config-if)# user-priority 3`

user-priority-regen-table	
Purpose	Set the value for the mapping of user-priority to regenerated user-priority.
Command Mode	Interface Configuration
Syntax	user-priority-regen-table user-priority <0-7> regenerated-user-priority <0-7>
Parameters	**user-priority**: Port priority that has to be mapped. **<0-7>**: User priority value. **regenerated-user-priority**: Regenerated values used for the user priority. **<0-7>**: Regenerated user priority value.
Example usage	`switch_a(config-if)# user-priority-regen-table user-priority 3 regenerateduser-priority 5`

RSTP Port Setting

bridge protocol rstp	
Purpose	Add an IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) bridge. After creating a bridge instance, add interfaces to the bridge using the bridge-group command. Bring the bridge instance into operation with the **no shutdown** command in Interface mode. Use the no parameter to remove the bridge.
Command Mode	Global Configuration
Syntax	bridge <1-32> protocol rstp (ring |) bridge <1-32> protocol rstp (vlan-bridge |)(ring |) no bridge <1-32>
Parameters	**<1-32>:**: Bridge group ID. **ring**: Add an RSTP bridge for a ring topology. **vlan-bridge**: Add a VLAN-aware bridge.
Example usage	`switch_a(config)# bridge 3 protocol rstp vlan-bridge`

bridge rapid-spanning-tree	
Purpose	Enable or disable RSTP on a specific bridge. Use the **bridge-forward** option with the no form of the command to place all ports on the specified bridge into the forwarding state.
Command Mode	Global Configuration
Syntax	bridge (<1-32> | backbone) rapid-spanning-tree enable no bridge <1-32> rapid-spanning-tree enable (bridge-forward |)
Parameters	**<1-32>:**: Bridge group ID. **enable**: Enable the spanning tree protocol. **bridge-forward**: Put all ports of specified bridge in forwarding state.
Example usage	`switch_a(config)# bridge 2 rapid-spanning-tree enable`
Note	When the **bridge-forward** option is not used with the no parameter, the default behavior puts all bridge ports in the blocking state.

MSTP Properties

bridge protocol mstp	
Purpose	Create a multiple spanning-tree protocol (MSTP) bridge of a specified parameter. This command creates an instance of the spanning tree and associates the VLANs specified with that instance. The MSTP bridges can have different spanning-tree topologies for different VLANs inside a region of “similar” MSTP bridges. The multiple spanning tree protocol, like the rapid spanning tree protocol, provides rapid reconfiguration capability, while providing load balancing ability. A bridge created with this command forms its own separate region unless it is added explicitly to a region using the region name command.
Command Mode	Global Configuration
Syntax	bridge <1-32> protocol mstp (ring |) no bridge <1-32>
Parameters	**<1-32>**: Bridge group ID. **ring**: (Optional) Enable rapid ring spanning-tree.
Example usage	`switch_a(config)# bridge 2 protocol mstp ring`

bridge multiple-spanning-tree	
Purpose	Enable MSTP on a bridge.
Command Mode	Global Configuration
Syntax	bridge <1-32> multiple-spanning-tree enable no bridge <1-32> multiple-spanning-tree enable (bridge-forward|)
Parameters	**<1-32>**: Bridge group ID. **enable**: Enable the spanning tree protocol. **bridge-forward**: Put all ports of specified bridge in forwarding state.
Example usage	`switch_a(config)# bridge 2 multiple-spanning-tree enable`
Note	When the **bridge-forward** option is not used with the no parameter, the default behavior puts all bridge ports in the blocking state.

MSTP Instance Setting

bridge max-hops	
Purpose	Set the maximum allowed hops for a BPDU in an MST region
Command Mode	Global Configuration
Syntax	[no] <1-32> max-hops <1-40>
Parameters	**<1-32>**: Bridge group ID **<1-40>**: Maximum hops for which the BPDU will be valid.
Example usage	`switch_a(config)# bridge 2 max-hops 25`
Note	Default maximum hops in an MST region are 20.

spanning-tree mst configuration	
Purpose	Enter the Multiple Spanning Tree Configuration mode.
Command Mode	Global Configuration
Syntax	spanning-tree mst configuration
Parameters	None
Example usage	`switch_a(config)# spanning-tree mst configuration` `switch_a(config-mst)#`

bridge instance	
Purpose	Add an MST instance to a bridge.
Command Mode	MST Configuration mode
Syntax	[no] bridge (<1-32> | backbone) instance (<1-63> | spbm | spbv)
Parameters	**<1-32>**: Bridge identifier. **backbone**: Backbone bridge. **<1-63>**: MST instance identifier. **spbm**: Shortest Path Bridging - MAC instance. **spbv**: Shortest Path Bridging - VID instance.
Example usage	`switch_a(config-mst)# bridge 4 instance 3`

bridge instance priority	
Purpose	Set the bridge instance priority. The lower the priority of the bridge, the better the chances is of the bridge becoming a root bridge or a designated bridge for the LAN.
Command Mode	Global Configuration
Syntax	bridge (<1-32> | backbone) instance <1-63> priority <0-61440> no bridge (<1-32> | backbone) instance <1-63> priority
Parameters	**<1-32>**: Bridge group ID **<1-63>**: The instance identifier. **<0-61440>**: Bridge priority value.
Example usage	`switch_a(config)# bridge 4 instance 3 priority 16384`
Note	The priority values can be set only in increments of 4096. The default value is 32768.

bridge instance vlan	
Purpose	Add multiple VLANs for the corresponding instance of a bridge.The VLANs must already be created. Use the no form to simultaneously remove multiple VLANs for the corresponding instance of a bridge.
Command Mode	MST Configuration mode
Syntax	bridge (<1-32> | backbone) instance (<1-63> | spbm | spbv) vlan VLANID no bridge (<1-32> | backbone) instance (<1-63> | spbm | spbv) vlan VLANID
Parameters	**<1-32>**: Bridge identifier. **backbone**: Backbone bridge. **<1-63>**: MST instance identifier. **spbm**: Shortest Path Bridging - MAC instance. **spbv**: Shortest Path Bridging - VID instance. **VLANID**: <2-4094>. Specify a single VLAN, a range, or a list.
Example usage	Associate VLANs 10 and 20 to instance 1 of bridge 1:
	`switch_a(config-mst)# bridge 1 instance 1 vlan 10,20`
	Add VLANs 10, 11, 12, 13, 14, and 15 to instance 1 of bridge 1:
	`switch_a(config-mst)# bridge 1 instance 1 vlan 10-15`
	Delete VLANs 10 and 11 from instance 1 of bridge 1:
	`switch_a(config-mst)# no bridge 1 instance 1 vlan 10,11`
Note	For a VLAN range, specify two VLAN identifiers: lowest then highest separated by hyphen. For a VLAN list, specify VLAN identifiers separated by commas. Do not enter spaces between the hyphens or commas.

instance vlan	
Purpose	Create an instance(s) of a VLAN for the default bridge (0). This command can be used only after the VLANs are defined; that is, LANs must be created before being associated with an MST instance (MSTI).
Command Mode	MST Configuration mode
Syntax	instance <1-63> vlan VLANID no instance <1-63> vlan VLANID
Parameters	**<1-63>**: MST instance identifier. **VLANID**: VLAN identifier(s)
Example usage	`switch_a(config-mst)# instance 2 vlan 30`

bridge region	
Purpose	Use this command to create an MST region and specify its name. MST bridges of a region form different spanning trees for different VLANs.
Command Mode	MST Configuration mode
Syntax	bridge <1-32> region REGION_NAME no bridge <1-32> region
Parameters	**<1-32>**: The bridge group ID. **REGION_NAME**: The name of the region.
Example usage	`switch_a(config-mst)# bridge 3 region IPI`
Note	By default, each MST bridge starts with the region name as its bridge address. This means each MST bridge is a region by itself, unless specifically added to one.

bridge revision	
Purpose	Specify the revision number, to be used for configuration information tracking.
Command Mode	MST Configuration mode
Syntax	bridge <1-32> revision <0-65535> no bridge <1-32>
Parameters	**<1-32>**: The bridge group ID **<0-65535>**: Revision number
Example usage	`switch_a(config-mst)# bridge 3 revision 25`
Note	The default value of revision number is 0.

region	
Purpose	Create an MST region of the default bridge, and specify a name to it.
Command Mode	MST Configuration mode
Syntax	[no] region REGION_NAME
Parameters	**REGION_NAME**: Name of the region
Example usage	`switch_a(config-mst)# region IPI`

9 VLAN

VLAN Information

show vlan	
Purpose	Show information for VLANs globally and per port.
Command Mode	Privileged exec
Syntax	show vlan brief show vlan private-valn show vlan classifier show vlan access-list show vlan access-map show vlan all show vlan auto show vlan filter show vlan static
Parameters	**None**
Example usage	`switch_a# show vlan brief`

VLAN Setting

vlan database	
Purpose	Enter the VLAN configuration mode
Command Mode	Global configuration
Syntax	vlan database
Parameters	**None**
Example usage	`switch_a(config)# vlan database`

vlan bridge	
Purpose	Add or remove a vlan under a bridge.
Command Mode	VLAN database
Syntax	vlan <1-3999> bridge<1-32> state [enable | disable] no vlan <2-4094> bridge <1-32>
Parameters	**<1-3999>:** VLAN ID **<1-32>:** Bridge ID
Example usage	`switch_a(config-vlan)# vlan 100 bridge 1 enable`

vlan mtu	
Purpose	Set the Maximum Transmission Unit (MTU) for a specified VLAN. Packets larger than the set MTU size are discarded.
Command Mode	VLAN database
Syntax	vlan <2-4094> mtu MTU_VAL vlan <2-4094> mtu MTU_VAL bridge <1-32> no vlan <2-4094> mtu no vlan <2-4094> mtu bridge <1-32>
Parameters	**MTU_VAL:** Value of the Maximum Transmission Unit **bridge:** Bridge group ID
Example usage	`switch_a(config-vlan)# vlan 2 mtu 1000 bridge 1`

vlan name	
Purpose	Enable or disable the name of a VLAN on the default bridge.
Command Mode	VLAN database
Syntax	vlan <2-4094> name WORD [state (enable | disable)]
Parameters	**enable:** Sets VLAN into enable state **disable:** Sets VLAN into disable state
Example usage	`switch_a(config-vlan)# vlan 100 name dilvish state enable`

Port Settings

switchport mode access	
Purpose	Set the interface to access mode, and classify untagged frames only. Received frames are classified based on the VLAN characteristics, then accepted or discarded based on the filtering criteria.
Command Mode	Interface Configuration
Syntax	switchport mode access [ingress-filter (enable | disable)]
Parameters	**ingress-filter**: Set the ingress filtering for the received frames. **enable**: Set the ingress filtering for received frames **disable**: Turn off ingress filtering to accept frames that do not meet the classification criteria. This is the default value.
Example usage	`switch_a(config-if)# switchport mode access ingress-filter enable`

switchport access	
Purpose	Change the default VLAN on the current interface.
Command Mode	Interface Configuration
Syntax	switchport access vlan VLAN_ID no switchport access vlan
Parameters	None
Example usage	`switch_a(config-if)# switchport access vlan 30`

switchport mode trunk	
Purpose	Set the interface as trunk, and specify only tagged frames.
Command Mode	Interface Configuration
Syntax	switchport mode trunk [ingress-filter (enable | disable)]
Parameters	**ingress-filter**: Set the ingress filtering for the received frames. **enable**: Set the ingress filtering for received frames. **disable**: Turn off ingress filtering to accept frames that do not meet the classification criteria. This is the default value.
Example usage	`switch_a(config-if)# switchport mode trunk ingress-filter enable`

switchport trunk allowed	
Purpose	Set the interface to trunk and add VLANs. For a VLAN range, specify two VLAN identifiers: the lowest and then the highest separated by a hyphen. For a VLAN list, specify the VLAN identifiers separated by commas. Do not enter spaces between the hyphens or commas. Use the no parameter to remove all VLAN identifiers configured on this port.
Command Mode	Interface Configuration
Syntax	switchport trunk allowed vlan all switchport trunk allowed vlan none switchport trunk allowed vlan add VLAN_ID switchport trunk allowed vlan except VLAN_ID switchport trunk allowed vlan remove VLAN_ID no switchport trunk
Parameters	**all**: Allow all VLANs to transmit and receive through the interface. **none**: Allow no VLANs to transmit and receive through the interface. **add**: Add these VLANs to the member set. **VLAN_ID**: VLAN identifier(s) <2-4094>. Specify a single VLAN, a VLAN range, or a VLAN list. **except**: All VLANs except these VLANs are part of the member set. **remove**: Remove these VLANs from the member set.
Example usage	`switch_a(config-if)# switchport trunk allowed vlan add V2`
Note	The result of not using this command is that ingress filtering is off, and that all frame types are classified and accepted.

switchport mode hybrid	
Purpose	Set the switching characteristics of the Layer 2 interface as hybrid, and classify both tagged and untagged frames. Set the interface acceptable frame types. This processing occurs after VLAN classification.
Command Mode	Interface Configuration
Syntax	switchport mode hybrid no switchport hybrid switchport mode hybrid acceptable-frame-type (all | vlan-tagged) switchport mode hybrid ingress-filter (enable | disable)
Parameters	None
Example usage	`switch_a(config-if)# switchport mode hybrid acceptable-frame-type all`

switchport hybrid	
Purpose	Set the switching characteristics of the Layer 2 interface as hybrid, and set the VLANs that will transmit/receive through the Layer 2 interface. Set the default VLAN for the interface.
Command Mode	Interface Configuration
Syntax	switchport hybrid allowed vlan all switchport hybrid allowed vlan none switchport hybrid allowed vlan except VLAN_ID switchport (hybrid) allowed vlan remove VLAN_ID switchport (hybrid) allowed vlan add VLAN_ID egress-tagged (enable|disable) no switchport hybrid switchport hybrid vlan VLAN_ID
Parameters	**all**: All VLANs can transmit and receive through the interface. **none**: No VLANs can transmit and receive through the interface. **except**: Allow all VLANs except these VLANs to transmit and receive through the interface. **VLAN_ID**: VLAN identifier <2-4094>. Specify a single VLAN, a VLAN range, or a VLAN list. **remove**: Remove these VLANs from the member set. **add**: Add these VLANs to the member set. **egress-tagged**: Tag outgoing frames. **enable**: Enable egress tagging for outgoing frames. **disable**: Disable egress tagging for outgoing frames.
Example usage	`switch_a(config-if)# switchport hybrid allowed vlan add 2 egress-tagged enable`

VLAN Translation

vlan translate	
Purpose	Turn VLAN translation on/off
Command Mode	Global Configuration
Syntax	vlan translate on vlan translate off
Parameters	None
Example usage	`switch_a(config-if)# vlan translate on`

switchport vlantrans	
Purpose	Set the ingress/egress VLAN translation from VLAN_ID to VLAN_ID
Command Mode	Interface Configuration
Syntax	switchport vlantrans ingress vlan VLAN_ID vlan VLAN_ID switchport vlantrans egress vlan VLAN_ID vlan VLAN_ID no switchport vlantrans ingress vlan VLAN_ID no switchport vlantrans egress vlan VLAN_ID
Parameters	**VLAN_ID**: VLAN ID
Example usage	`switch_a(config-if)# no switchport vlantrans ingress vlan 20`

Private VLAN

private-vlan primary	
Purpose	Create a primary VLAN.
Command Mode	VLAN Configuration
Syntax	[no] private-vlan <2-4094> primary bridge <1-32>
Parameters	**<2-4094>:** Private VLAN identifier. **<1-32>**: Bridge ID
Example usage	`switch_a(config-vlan)# private-vlan 2 primary bridge 1`

private-vlan association	
Purpose	Associate a secondary VLAN to a primary VLAN. Only one isolated VLAN can be associated to a primary VLAN. Multiple community VLANs can be associated to a primary VLAN.
Command Mode	VLAN Configuration
Syntax	private-vlan <2-4094> association add VLAN_ID bridge <1-32> private-vlan <2-4094> association remove VLAN_ID bridge <1-32> no private-vlan <2-4094> association bridge <1-32>
Parameters	**<2-4094>**: Private VLAN identifier. **add**: Add values associated with a single VLAN. **remove**: Remove values associated with a single VLAN. **VLAN_ID**: Secondary VLAN identifier <2-4094>. **<1-32>**: Bridge group ID.
Example usage	`switch_a(config-vlan)# private-vlan 2 association add 3-4 bridge`

private-vlan isolated	
Purpose	Create an isolated VLAN. Use the no form to remove the specified private VLAN.
Command Mode	VLAN Configuration
Syntax	private-vlan <2-4094> isolated bridge <1-32> no private-vlan <2-4094> isolated bridge <1-32>
Parameters	**<2-4094>**: Private VLAN identifier. **<1-32>**: Bridge identifier.
Example usage	`switch_a(config-vlan)# private-vlan 3 isolated bridge 1`

private-vlan community	
Purpose	Set a VLAN type for a private (community) VLAN. Use the no form to remove the specified private VLAN.
Command Mode	VLAN Configuration
Syntax	private-vlan <2-4094> community bridge <1-32> no private-vlan <2-4094> community bridge <1-32>
Parameters	**<2-4094>**: Private VLAN identifier. **<1-32>**: Bridge identifier.
Example usage	`switch_a(config-vlan)# private-vlan 4 community bridge 1`

switchport mode private-vlan	
Purpose	Make a Layer 2 port a host port or promiscuous port.
Command Mode	Interface Configuration
Syntax	[no] switchport mode private-vlan (host | promiscuous)
Parameters	**host**: Port can communicate with all other host ports assigned to the same community VLAN, but it cannot communicate with the ports in the same isolated VLAN. All communications outside of this VLAN must pass through a promiscuous port in the associated primary VLAN. **promiscuous**: Port can communicate with all interfaces, including the community and isolated ports within a private VLAN
Example usage	`switch_a(config-if)# switchport mode private-vlan promiscuous`

switchport private-vlan host-association	
Purpose	Associate a primary VLAN and a secondary VLAN to a host port. Only one primary and secondary VLAN can be associated to a host port.
Command Mode	Interface Configuration
Syntax	switchport private-vlan host-association <2-4094> add <2-4094> no switchport private-vlan host-association
Parameters	**<2-4094>**: VLAN identifier of the primary VLAN. **add**: Add the secondary VLAN. **<2-4094>**: VLAN identifier of the secondary VLAN (either isolated or community).
Example usage	`switch_a(config-if)# switchport private-vlan host-association 2 add 3`

MAC/Subnet/Protocol Based VLAN

vlan classifier rule ipv4	
Purpose	Create a subnet-based VLAN classifier rule and map it to a specific VLAN. If the source IP address matches the IP subnet specified in the VLAN classifier rule, received packets are mapped to the designated VLAN.
Command Mode	Global Configuration
Syntax	vlan classifier rule <1-256> ipv4 <A.B.C.D/M> vlan <2-4094> no vlan classifier rule <1-256> ipv4 <A.B.C.D/M>
Parameters	**A.B.C.D/M**: The IPv4 address classification in A.B.C.D/M format. **vlan**: The VLAN to which an untagged packet is mapped **<2-4094>**: VLAN ID
Example usage	`switch_a(config)# vlan classifier rule 2 ipv4 20.20.20.2/24 vlan 2`

vlan classifier rule mac	
Purpose	Create a subnet-based VLAN classifier rule and map it to a specific VLAN. If the source IP address matches the IP subnet specified in the VLAN classifier rule, received packets are mapped to the designated VLAN.
Command Mode	Global Configuration
Syntax	vlan classifier rule <1-256> mac WORD vlan <2-4094> no vlan classifier rule <1-256>
Parameters	**WORD**: Mac address classification. Enter the address in HHHH.HHHH.HHHH format. **vlan**: The VLAN to which an untagged packet is mapped <2-4094>.
Example usage	`switch_a(config)# vlan classifier rule 2 mac fe80::22e::b5ff:fee8:6/64 vlan 2`

vlan classifier rule proto	
Purpose	Create a subnet-based VLAN classifier rule for a protocol and map it to a specific VLAN. If the source IP address matches the IP subnet specified in the VLAN classifier rule, received packets are mapped to the designated VLAN.
Command Mode	Global Configuration
Syntax	vlan classifier rule <1-256> proto (ip | ipv6 | ipx | x25 | arp | rarp | atalkddp | atalkaarp | atmmulti | atmtransport | pppdiscovery | pppsession | xeroxpup | xeroxaddrtrans | g8bpqx25 | ieeepup | ieeeaddrtrans | dec | decdnadumpload | decdnaremoteconsole | decdnarouting | declat | decdiagnostics | deccustom | decsyscomm | <0-65535>) encap (ethv2 | snapllc | nosnapllc) (vlan <2-4094>|) no vlan classifier rule <1-256>
Parameters	**<0-65535>**: Ethernet decimal **arp**: Address Resolution Protocol **atalkaarp**: Appletalk AARP **atalkddp**: Appletalk DDP **atmmulti**: MultiProtocol Over ATM **atmtransport**: Frame-based ATM Transport **dec**: DEC Assigned **deccustom**: DEC Customer use **decdiagnostics**: EC Diagnostics **decdnadumpload**: DEC DNA Dump/Load **decdnaremoteconsole**: DEC DNA Remote Console **decdnarouting**: DEC DNA Routing **declat**: DEC LAT **decsyscomm**: DEC Systems Comms Arch **g8bpqx25**: G8BPQ AX.25 **ieeeaddrtrans**: Xerox IEEE802.3 PUP Address Translation **ieeepup**: Xerox IEEE802.3 PUP

	ip: IP address **ipv6**: IPv6 address **ipx**: IPX address **pppdiscovery**: PPPoE discovery **pppsession**: PPPoE session **rarp**: Reverse Address Resolution **x25**: CCITT X.25 **xeroxaddrtrans**: Xerox PUP Address Translation **Xerox**: PUP **encap**: packet encapsulation **ethv2**: Ethernet v2 **nosnapllc**: LLC without snap encapsulation **snapllc**: LLC snap encapsulation **vlan**: The VLAN to which an untagged packet is mapped <2-4094>.
Example usage	`switch_a(config)# vlan classifier rule 2 proto ip encap ethv2 vlan 2`

vlan classifier group	
Purpose	Create a subnet-based VLAN classifier group. A group indicates a VLAN classifier group ID.
Command Mode	Global Configuration
Syntax	vlan classifier group <1-16> (add | delete) rule <1-256> no vlan classifier group <1-16>
Parameters	**add**: Adds a rule to a group. **delete**: Deletes a rule from a group. **rule**: Indicates the VLAN classifier rule identifier <1-256>.
Example usage	`switch_a(config)# vlan classifier group 1 delete rule 1`

vlan classifier activate	
Purpose	Activate VLAN classifier
Command Mode	Interface Configuration
Syntax	vlan classifier activate <1-16> vlan <1-3999> no vlan classifier activate <1-16>
Parameters	**add**: Adds a rule to a group. **delete**: Deletes a rule from a group. **rule**: Indicates the VLAN classifier rule identifier <1-256>.
Example usage	`switch_a(config-if)# vlan classifier activate 1 vlan 100`

10 QOS

Global Configuration

show mls qos	
Purpose	Display various QoS configuration and statistics.
Command Mode	Privileged Exec
Syntax	show mls qos show mls qos aggregate-policer <NAME> show mls qos cosq-stats <IFNAME> <0-7> show mls qos interface <IFNAME> show mls qos map dscp-queue show qos tail-drop counters <IFNAME> <0-7> show user-priority interface <IFNAME>
Parameters	**NAME**: aggregator policer name **<0-7>**: queue number
Example usage	`switch_a# show mls qos`

mls qos	
Purpose	Enable/disable QoS
Command Mode	Global Configuration
Syntax	mls qos enable no mls qos
Parameters	None
Example usage	`switch_a(config)# mls qos enable`

mls qos aggregate-police	
Purpose	Specify policer parameters to apply to multiple traffic classes in the same policy map.
Command Mode	Global Configuration
Syntax	mls qos aggregate-police NAME <1-1000000> <1-20000> no mls qos aggregate-police NAME
Parameters	**NAME**: Name of the aggregate policer. **<1-1000000>**: Average traffic rate in bits per second (kbps). **<1-20000>**: Normal burst size in kilobytes (bytes).
Example usage	`switch_a(config)# mls qos aggregate-police transmit1 48000 8000`

mls qos cos-queue	
Purpose	Configure CoS (Class of Service) queue.
Command Mode	Global Configuration
Syntax	mls qos cos-queue <0-7> <0-3> no mls qos cos-queue <0-7> <0-3>
Parameters	**<0-7>:** Priority for the queue. **<0-3>**: Queue identifier
Example usage	`switch_a(config)# mls qos cos-queue 5 3`

priority-queue	
Purpose	Set the egress expedit queue or weighted deficit round robin.
Command Mode	Global Configuration
Syntax	priority-queue strict priority-queue wdrr no priority-queue out
Parameters	**strict**: egress expedite queue **wdrr**: weighted deficit round robin
Example usage	`switch_a(config)#priority-queue strict`

wrr-queue bandwidth	
Purpose	Specify the bandwidth ratios of the transmit queues.
Command Mode	Global Configuration
Syntax	wrr-queue bandwidth <1-127> <1-127> <1-127> <1-127> <1-127> <1-127> <1-127> <1-127> no wrr-queue bandwidth
Parameters	**<1-127>**: Weight of queues 0-7. **<0-7>**: QoS queue ID.
Example usage	`switch_a(config)# wrr-queue bandwidth 1 30 40 20 60 80 70 100`

wrr-queue cos-map	
Purpose	Specify CoS values for a queue.
Command Mode	Global Configuration
Syntax	wrr-queue cos-map <0-7> (<0-7> | <0-7> <0-7> | <0-7> <0-7> <0-7> | <0-7> <0-7> <0-7> <0-7> | <0-7> <0-7> <0-7> <0-7> <0-7> | <0-7> <0-7> <0-7> <0-7> <0-7> <0-7> | <0-7> <0-7> <0-7> <0-7> <0-7> <0-7> <0-7> | <0-7> <0-7> <0-7> <0-7> <0-7> <0-7> <0-7> <0-7>) no wrr-queue cos-map <0-7>
Parameters	**<0-7>**: Queue identifier. **<0-7>**: 1-8 CoS values separated by spaces.
Example usage	`switch_a(config)# wrr-queue bandwidth 1 30 40 20 60 80 70 100`

DSCP

mls qos map dscp-queue	
Purpose	Set dscp to queue
Command Mode	Global Configuration
Syntax	mls qos map dscp-queue <0-63> <0-7> no mls qos map dscp-queue
Parameters	**<0-63>**: dscp value. **<0-7>**: CoS Queue ID
Example usage	`switch_a(config)# mls qos map dscp-queue 10 5`

Interface

tail-drop threshold	
Purpose	Configure the tail-drop threshold percentages for a queue. Use the no parameter return to the default setting.
Command Mode	Interface Configuration
Syntax	tail-drop threshold <0-7> <1-100> <1-100>
Parameters	**<0-7>** Queue identifier. **<1-100>** Minimum threshold percentage. **<1-100>** Maximum threshold percentage.
Example usage	`switch_a(config-if)# tail-drop threshold 1 60 1005`

11 Access Control Lists (ACL)

ACL Information

show class-map	
Purpose	Display the QoS class maps to define the match criteria to classify traffic.
Command Mode	Privileged Exec
Syntax	show class-map NAME
Parameters	**NAME**: Name of the class map
Example usage	`switch_a# show class-map cmap10`

show policy-map	
Purpose	Display the QoS class maps to define the match criteria to classify traffic.
Command Mode	Privileged Exec
Syntax	show policy-map show policy-map NAME
Parameters	**NAME**: Name of the policy map
Example usage	`switch_a# show policy-map pmap10`

show access-lists	
Purpose	Display current access lists.
Command Mode	Privileged Exec
Syntax	show access-lists
Parameters	None
Example usage	`switch_a# show access-lists`

show ip access-lists	
Purpose	Display contents of all current IP access lists.
Command Mode	Privileged Exec
Syntax	show ip-access-lists [<1-99> | <100-199> | <1300-1999> | <2000-2699> | WORD]
Parameters	**<1-99>**: Standard access list **<100-199>**: Extended access list **<1300-1999>**: Standard access list (expanded range) **<2000-2699>**: Extended access list (expanded range) **WORD**: Access list name
Example usage	`switch_a# show ip access-lists 50`

show qos-access-list	
Purpose	Display IP and MAC ACLs.
Command Mode	Privileged Exec
Syntax	show qos-access-list show qos-access-list [<1-99> | <100-199> | <1300-1999> | <2000-2699> | WORD]
Parameters	**<1-99>**: Standard access list **<100-199>**: Extended access list **<1300-1999>**: Standard access list (expanded range) **<2000-2699>**: Extended access list (expanded range) **WORD**: Access list name
Example usage	`switch_a# show qos-access-list 1`

ACL Configuration

access-list	
Purpose	Add an access list entry.
Command Mode	Global Configuration
Syntax	[no] access-list (<1-99> | <1300-1999>) (deny | permit) any [no] access-list (<1-99> | <1300-1999>) (deny | permit) host <A.B.C.D> [no] access-list (<1-99> | <1300-1999>) (deny | permit) <A.B.C.D> WILD [no] access-list (<1-99> | <1300-1999>) remark <LINE>
Parameters	**<1-99>**: IP standard access list **<1300-1999>**: IP standard access list (expanded range). **deny**: Route to reject. **permit**: Route to permit. **A.B.C.D**: IP address to match **WILD**: Wildcard bits **any**: Allows any IP address or prefix to match. **remark**: Access list entry comment. **LINE**: Multi-line, access-list entry comment up to 100 characters.
Example usage	`switch_a(config)# access-list 67 deny 1.1.1.0 0.0.0.255`

ip-access-list (std)	
Purpose	Create a standard IP access-control list (ACL). Use the no parameter to delete the ACL. Enable QoS globally before using this command. Use the no parameter to delete the ACL.
Command Mode	Global Configuration
Syntax	[no] ip-access-list (<1-99> | <1300-1999>) (deny | permit) A.B.C.D [no] ip-access-list (<1-99> | <1300-1999>) (deny | permit) A.B.C.D WILD [no] ip-access-list (<1-99> | <1300-1999>) (deny | permit) [any | host <A.B.C.D>]
Parameters	**<1-99>**: IP standard ACL. **<100-199>**: IP extended ACL. **deny**: Deny traffic if conditions matched. **permit**: Permit traffic if conditions matched. **A.B.C.D**: Address to match. **WILD**: Wildcard bits. **any**: Any source host. **host**: A single source host for extended ACLs.
Example usage	`switch_a(config)#ip-access-list 1 permit 192.5.255.0 0.0.0.255`

ip-access-list (extd)	
Purpose	Create an extended IP access-control list (ACL). Use the no parameter to delete the ACL. Enable QoS globally first.
Command Mode	Global Configuration
Syntax	[no] ip-access-list (<100-199> | <2000-2699>) (deny | permit) (PROTOCOL) any [any | host <A.B.C.D> | <A.B.C.D> <A.B.C.D>] [no] ip-access-list (<100-199>|<2000-2699>) (deny | permit) (PROTOCOL) host [<A.B.C.D> any | <A.B.C.D> host <A.B.C.D> | <A.B.C.D> WILD <A.B.C.D>] [no] ip-access-list (<100-199>|<2000-2699>) (deny | permit) (PROTOCOL) <A.B.C.D> WILD any [no] ip-access-list (<100-199>|<2000-2699>) (deny | permit) (PROTOCOL) <A.B.C.D> WILD host <A.B.C.D> [no] ip-access-list (<100-199>|<2000-2699>) (deny | permit) (PROTOCOL) <A.B.C.D> WILD <A.B.C.D> WILD
Parameters	**<100-199>**: Range for IP extended ACL. **<2000-2699>**: Range for IP extended access list (expanded range). **PROTOCOL**: ip, udp, tcp, gre, igmp, pim, rsvp, ospf, vrrp, ipcomp, any, <0-255> (IANA assigned protocol). **deny**: Deny traffic if conditions matched. **permit**: Permit traffic if conditions matched. **A.B.C.D**: Source / destination to match. **WILD**: Wildcard bits in A.B.C.D format. **any**: Any source host. **host**: A single source host for extended ACLs.
Example usage	`ip-access-list 199 deny gre any any`

ip-access-list standard	
Purpose	Create a named standard IP access-control list (ACL). Use the no parameter to delete the ACL. Enable QoS globally first.
Command Mode	Global Configuration
Syntax	[no] ip-access-list standard <NAME> (deny | permit) any [no] ip-access-list standard <NAME> (deny | permit) <A.B.C.D> [no] ip-access-list standard <NAME> (deny | permit) <A.B.C.D> WILD
Parameters	**NAME**: Name of standard ACL. **deny**: Deny certain traffic if conditions matched. **permit**: Permit certain traffic if conditions matched. **A.B.C.D**: Address to match. **WILD**: Wildcard bits in A.B.C.D format. **any**: Any source host.
Example usage	`switch_a(config)#ip-access-list standard test2 permit 192.5.255.0 0.0.0.255`

ip-access-list extended	
Purpose	Create a named extended IP access-control list (ACL). Enable QoS globally first. Use no parameter to delete ACL.
Command Mode	Global Configuration
Syntax	[no] ip-access-list extended <NAME> (deny | permit) (PROTOCOL) any any [no] ip-access-list extended <NAME> (deny | permit) (PROTOCOL) any host <A.B.C.D> [no] ip-access-list extended <NAME> (deny | permit) (PROTOCOL) any <A.B.C.D> WILD [no] ip-access-list extended <NAME> (deny | permit) (PROTOCOL) host <A.B.C.D> any [no] ip-access-list extended <NAME> (deny | permit) (PROTOCOL) host <A.B.C.D> host <A.B.C.D> [no] ip-access-list extended <NAME> (deny | permit) (PROTOCOL) host <A.B.C.D> WILD <A.B.C.D> [no] ip-access-list extended <NAME> (deny | permit) (PROTOCOL) <A.B.C.D> WILD any [no] ip-access-list extended <NAME> (deny | permit) (PROTOCOL) <A.B.C.D> WILD host <A.B.C.D> [no] ip-access-list extended <NAME> (deny | permit) (PROTOCOL) <A.B.C.D> WILD <A.B.C.D> WILD
Parameters	**NAME**: Name of extended ACL. **PROTOCOL**: ip, udp, tcp, gre, igmp, pim, rsvp, ospf, vrrp, ipcomp, any, <0-255> (IANA assigned protocol). **deny**: Deny traffic if conditions matched. **permit**: Permit traffic if conditions matched. **A.B.C.D**: Source / destination address. **WILD**: Wildcard bits in A.B.C.D format. **any**: Any source host. **host**: A single source host for extended ACLs.
Example usage	`switch_a(config)# ip-access-list extended 2001 permit ip 192.5.255.0 0.0.0.255 any`

mac-access-list	
Purpose	Create a MAC access-control list (ACL). Use the no parameter to delete the ACL. Enable QoS globally before using this command.
Command Mode	Global Configuration
Syntax	**MAC ACL for any source and a destination specified by the MAC and MASK parameters:** mac-access-list <2000-2699> (deny | permit) any MAC MASK <1-8> **MAC ACL for a source specified by the MAC and MASK parameters and any destination:** mac-access-list <2000-2699> (deny | permit) MAC MASK any <1-8> **MAC ACL for a source specified by the first MAC and MASK parameters and a destination specified by the second MAC and MASK parameters:** mac-access-list <2000-2699> (deny | permit) MAC MASK MAC MASK <1-8> **MAC ACL for a source or destination specified by the MAC parameter:** mac-access-list <2000-2699> (source | destination) MAC priority <0-7> no mac-access-list <2000-2699> (deny | permit) any MAC MASK <1-8> no mac-access-list <2000-2699> (deny | permit) MAC MASK any <1-8> no mac-access-list <2000-2699> (deny | permit) MAC MASK MAC MASK <1-8> no mac-access-list <2000-2699> (source | destination) MAC priority <0-7>
Parameters	**<2000-2699>**: Expanded range for IP extended ACL. **deny**: Deny certain traffic if conditions match. **permit**: Permit certain traffic if conditions match. **any**: Any source or destination. **MAC**: MAC address; in HHHH.HHHH.HHHH format. **MASK**: Part of the MAC address to ignore in hexadecimal format. **<1-8>**: Packet format: 1 Ethernet II, 2 802.3, 4 SNAP, 8 LLC **source**: Packets with source MAC address **destination**: Packets with destination MAC address. **priority**: Priority class. **<0-7>**: Priority value.
Example usage	`switch_a(config)# mac-access-list 2002 permit 2222.2222.2222 8 any 2`

class-map	
Purpose	Create a class map. Enable QoS globally before using this command. Use the no parameter to delete the class map. This command will put the switch into Class Map Configuration mode
Command Mode	Global Configuration
Syntax	[no] class-map NAME
Parameters	**NAME**: Name of class map
Example usage	`switch_a(config)# class-map ahax` `switch_a(config-cmap)#`

match access-group	
Purpose	Define match criterion for a class map. Enable QoS globally before using this command.
Command Mode	Class Map Configuration
Syntax	[no] match access-group (<1-199> | <1300-2699> | WORD)
Parameters	**<1-99>**: Number of standard ACL **<1300-2699>**: Number of extended ACL **WORD**: Name of the ACL

Example usage	Configure a class map named test10 with 1 match criterion: access list 103, which allows traffic from any source to any destination.
	`switch_a(config)#ip-access-list 103 permit ip any any`
	`switch_a(config)#class-map test10`
	`switch_a(config-cmap)#match access-group 103`

match cos	
Purpose	Match packetS based on class of service (CoS). Enable QoS first.
Command Mode	Class Map Configuration
Syntax	match cos (<0-7> | <0-7> | <0-7> | <0-7> | <0-7> | <0-7> | <0-7> | <0-7>) no match cos match cos inner (<0-7> | <0-7> | <0-7> | <0-7> | <0-7> | <0-7> | <0-7> | <0-7>) no match cos inner
Parameters	**<0-7>**: IEEE 802.1Q/ISL CoS value **inner**: Match the inner cos of QinQ packets
Example usage	`switch_a(config-cmap)# match cos 7`

match ip-dscp	
Purpose	Define the list to match against incoming packets. Enable QoS globally before using this command. Use the no parameter to disable this configuration.
Command Mode	Class Map Configuration
Syntax	match ip-dscp (<0-63> | <0-63> <0-63> | <0-63> <0-63> <0-63> | <0-63> <0-63> <0-63> <0-63> | <0-63> <0-63> <0-63> <0-63> <0-63> | <0-63> <0-63><0-63> <0-63> <0-63> <0-63> | <0-63> <0-63> <0-63> <0-63> <0-63> <0-63> <0-63> | <0-63> <0-63> <0-63> <0-63> <0-63> <0-63> <0-63> <0-63>) no match ip-dscp
Parameters	**<0-63>**: 1-8 IP DSCP values separated by a space.
Example usage	`switch_a(config-cmap)# match ip-dscp 56`

match ip-precedence	
Purpose	Identify IP precedence values as match criteria. Use the no parameter to remove IP precedence values from a class map.
Command Mode	Class Map Configuration
Syntax	match ip-precedence (<0-7> | <0-7> <0-7> | <0-7> <0-7> <0-7> | <0-7> <0-7> <0-7> <0-7> | <0-7> <0-7> <0-7> <0-7> <0-7> | <0-7> <0-7> <0-7> <0-7><0-7> <0-7> | <0-7> <0-7> <0-7> <0-7> <0-7> <0-7> <0-7> | <0-7> <0-7> <0-7> <0-7> <0-7> <0-7> <0-7> <0-7>) no match ip-precedence <0-7>
Parameters	**<0-7>**: 1-8 precedence values separated by a space.
Example usage	`switch_a(config-cmap)# match ip-precedence 5 6 4 3`

match layer4	
Purpose	Identify UDP or TCP ports as the match criteria. Use the no parameter to remove match criteria.
Command Mode	Class Map Configuration
Syntax	match layer4 (any | tcp | udp) (source-port | destination-port) <1-65535> no match layer4 (any | tcp | udp) (source-port | destination-port) <1-65535> match layer4-range (any | tcp | udp) (source-port | destination-port) <1-65535> to <1-65535> no match layer4-range
Parameters	**source-port**: Source UDP or TCP port. **destination-port**: Destination UDP or TCP port.
Example usage	`switch_a(config-cmap)# match layer4 source-port 20`

match mpls exp-bit topmost	
Purpose	Define the match criterion of the MPLS (Multiprotocol Label Switching) experimental bit value in the topmost label for a class map. Use the no parameter to remove this criterion from a class map.
Command Mode	Class Map Configuration
Syntax	match mpls exp-bit topmost (<0-7> | <0-7> | <0-7> | <0-7> | <0-7> | <0-7> | <0-7> | <0-7>) no match mpls exp-bit topmost
Parameters	**<0-7>**: 1-8 experimental values separated by a space.
Example usage	`switch_a(config-cmap)# match mpls exp-bit topmost 0 1 2 3 4 5 6 7`

match traffic-type	
Purpose	Set the match criteria according to traffic type.
Command Mode	Class Map Configuration
Syntax	match traffic-type (TYPE) (traffic-type-and-queue | traffic-type-or-queue) no match traffic-type [TYPE]
Parameters	**TYPE**: all, arp, broadcast, management, multicast, non-tcp-udp, queue0, queue1, queue2, queue3, tcp-control, tcp-data, udp, unicast, unknown-multicast, unknown-unicast
Example usage	`switch_a(config-cmap)# match traffic-type all`

match vlan	
Purpose	Define the VLAN identifier or a range of VLANS used as a match criteria to classify traffic. Use the no parameter to disable the VLAN ID used as match criteria.
Command Mode	Class Map Configuration
Syntax	match vlan <1-3999> no match vlan match vlan inner <1-3999> no match vlan inner match vlan-range <1-3999> to <1-3999> no match vlan-range match vlan-range inner <1-3999> no match vlan-range inner
Parameters	**<1-3999>**: VLAN identifier
Example usage	`switch_a(config-cmap)# match vlan 1000`

policy-map	
Purpose	Create a policy map. Use the no parameter to delete an existing policy map. This command will put the switch into Policy Map Configuration mode.
Command Mode	Global Configuration
Syntax	[no] policy-map NAME
Parameters	**NAME**: name of the policy map
Example usage	`switch_a(config)# policy-map groo` `switch_a(config-pmap)#`

class	
Purpose	Define a traffic classification. Enable QoS globally before using this command. Use the no parameter to delete an existing class-map. Using this command will put the switch into Policy Map Class configuration mode (pmap-c).
Command Mode	Policy Map Configuration
Syntax	[no] class NAME
Parameters	**NAME**: name of the class map
Example usage	`switch_a(config-pmap)# class arcadio` `switch_a(config-pmap-c)#`

set cos	
Purpose	Set a CoS value to assign to classified traffic or enable copying the priority bit (pbit) from the inner VLAN to the outer VLAN based on policy. Use the no parameter to remove a CoS value, or disable pbit copying.
Command Mode	Policy Map Class Configuration
Syntax	set cos <0-7> set cos <0-7> (map | remark) set cos <0-7> cos-inner (map | remark) no set cos set cos-inner (<0-7> | cos) no set cos-inner
Parameters	**<0-7>**: CoS value to assign to classified traffic. **cos-inner**: Copy pbit from the inner VLAN to the outer VLAN based on policy. **map**: Map to cos value (default) **remark**: Remark to cos value
Example usage	`switch_a(config-cmap-c)# set cos 2`

set drr-priority	
Purpose	Set a deficit round-robin (DRR) priority. Use the no parameter to remove this priority.
Command Mode	Policy Map Class Configuration
Syntax	set drr-priority <0-7> quantum <1-255> no set drr-priority
Parameters	**<0-7>**: CoS value to assign to classified traffic. **<1-255>**: DRR quantum value.
Example usage	`switch_a(config-cmap-c)# set drr-priority 1 quantum 1`

set ip-dscp	
Purpose	Set a DSCP value to assign to classified traffic. Use the no parameter to remove a DSCP value.
Command Mode	Policy Map Class Configuration
Syntax	set ip-dscp <0-63> no set ip-dscp
Parameters	**<0-63>**: DSCP value.
Example usage	`switch_a(config-cmap-c)# set ip-dscp 40`

set ip-precedence	
Purpose	Set an IP-precedence value to assign to classified traffic. Use the no parameter to remove an IP-precedence value.
Command Mode	Policy Map Class Configuration
Syntax	set ip-precedence <0-7> no set ip-precedence
Parameters	**<0-7>**: IP precedence value
Example usage	`switch_a(config-cmap-c)# set ip-precedence 2`

set mirror-to-port	
Purpose	Set a new value in the packet redirect packet to new interface (interface name).
Command Mode	Policy Map Class Configuration
Syntax	set mirror-to-port <IFNAME> no set mirror-to-port
Parameters	**IFNAME**: Interface name
Example usage	`switch_a(config-cmap-c)# set mirror-to-port ge7`

set mpls exp-bit topmost	
Purpose	Set the MPLS experimental-bit value in the topmost label for a policy map. Set a new MPLS experimental-bit value in a packet to classify MPLS traffic. Use the no parameter to remove this setting from a policy map.
Command Mode	Policy Map Class Configuration
Syntax	set mpls exp-bit topmost <0-7> no set mpls exp-bit topmost
Parameters	**<0-7>**: Experimental value.
Example usage	`switch_a(config-cmap-c)# set mpls exp-bit topmost 7`

set redirect-to-port	
Purpose	Set a new value in the packet redirect packet to new interface (interface name).
Command Mode	Policy Map Class Configuration
Syntax	set redirect-to-port <IFNAME> no set redirect-to-port
Parameters	**IFNAME**: Interface name
Example usage	`switch_a(config-cmap-c)# set redirect-to-port ge1`

set vlan	
Purpose	Set a new value in the packet VLAN (VLAN value), or set a new value in the packet VLAN (VLAN value) (CoS value)
Command Mode	Policy Map Class Configuration
Syntax	set vlan <1-3999> set vlan <1-3999> <0-7> no set vlan
Parameters	**<1-3999>**: new VLAN value **<0-7>**: CoS value.
Example usage	`switch_a(config-cmap-c)# set vlan 1000 6`

set vlan-priority	
Purpose	Set a VLAN priority for the queues.
Command Mode	Policy Map Class Configuration
Syntax	set vlan-priority <0-7> <0-7> <0-7> <0-7> <0-7> <0-7> <0-7> <0-7> no set vlan-priority
Parameters	**<0-7>**: Priorities for queues.
Example usage	`switch_a(config-cmap-c)# set vlan-priority 1 1 1 1 1 1 1 1`

mode	
Purpose	Set operation mode (allow accept flow traffic) (deny drop flow traffic).
Command Mode	Policy Map Class Configuration
Syntax	mode (allow|deny) no mode (allow|deny)
Parameters	**allow**: Allow accept flow traffic (default mode) **deny**: Deny drop flow traffic
Example usage	`switch_a(config-cmap-c)# mode deny`

police	
Purpose	Specify a Single Rate Three Color Marker (srTCM) or Two Rate Three Color Marker (trTCM) policer.
Command Mode	Policy Map Class Configuration
Syntax	police [srtcm | trtcm]<1-1000000> <1-20000> <1-20000> exceed-action (drop | flow-control) resetflow-control-mode available-bucket-room (full | cbs) no police [srtcm | trtcm] <1-1000000> <0-20000000> <1-20000000> exceed-action (drop | flow-control) reset-flow-control-mode available-bucket-room (full | cbs)
Parameters	**<1-1000000>**: Average traffic rate in kbps. **<1-20000>**: Burst size in kbps. **<1-20000>**: Exceed burst size in kbps. **exceed-action**: Action when rates are exceeded. **drop**: Drop the frame. **flow-control**: Send a pause frame and pass the packet. **reset-flow-control-mode**: Generate flow control. **available-bucket-room**: When to de-assert flow control. **full**: When bucket room is full. **cbs**: When bucket has enough room.
Example usage	`switch_a(config-cmap-c)# police 48000 8000 exceed-action drop`

police-aggregate	
Purpose	Set add an access list entry standard access list (Address to match) (Wildcard bits)
Command Mode	Policy Map Class Configuration
Syntax	police-aggregate <NAME> no police-aggregate <NAME>
Parameters	**NAME**: Aggregate policer name
Example usage	`switch_a(config-cmap-c)# police-aggregate grativo`

policing meter	
Purpose	Set a policer meter for the classified traffic average traffic rate by burst rate(policing ratio).
Command Mode	Policy Map Class Configuration
Syntax	policing meter <1-255> no policing meter
Parameters	**<1-255>**: policing ratio
Example usage	`switch_a(config-cmap-c)# policing meter 100`

service-policy input/output	
Purpose	Apply a policy map to the input or output of an interface.
Command Mode	Interface Configuration
Syntax	service-policy input <NAME> no service-policy input <NAME> service-policy output <NAME> no service-policy output <NAME>
Parameters	**NAME**: Policy input or output name
Example usage	`switch_a(config-if)# service-policy input pmap1`

12 SNMP (Simple Network Management Protocol)

SNMP Configuration

snmp-server	
Purpose	Configure an SNMP server.
Command Mode	Global Configuration
Syntax	[no] snmp-server enable snmp-server community (get | set) WORD snmp-server contact WORD snmp-server location WORD snmp-server description WORD snmp-server mac-nofitication (history-size | interval) <1-65535> snmp-server trap-community <1-5> LINE snmp-server trap-ipaddr <1-5> <A.B.C.D> [no] snmp-server trap-type enable [linkdown | linkup | mac-notification]
Parameters	**WORD**: SNMP community name **WORD**: Contact string **WORD**: Location string **WORD**: Description string **history size**: Maximum number of entries in the MAC notification history table **interval**: the notification trap interval in seconds between each set of traps that are generated <1-65535> **LINE**: Community name
Example usage	`switch_a(config)# snmp-server trap-community 1 Trap_Group_1`

snmp-server trap mac-notification	
Purpose	Enable SNMP traps for MAC-notification events.
Command Mode	Interface Configuration
Syntax	[no] snmp-server trap mac-notification
Parameters	None
Example usage	`switch_a(config-if)# snmp-server trap mac-notification`

snmp v3-user	
Purpose	Configure SNMP version 3 user
Command Mode	Global Configuration
Syntax	snmp v3-user WORD (ro | rw) (auth | noauth) ((md5 | sha) WORD|) snmp v3-user WORD (ro | rw) priv (md5 | sha) WORD des WORD no snmp v3-user WORD
Parameters	None
Example usage	`switch_a(config-if)# snmp v3-user test ro auth`

13 IEEE 802.1X

802.1x Information

show dot1x	
Purpose	Display dot1x configuration and settings
Command Mode	Privileged Exec
Syntax	show dot1x show dot1x all show dot1x diagnostics interface IFNAME show dot1x interface IFNAME show dot1x sessionstatistics interface IFNAME show dot1x statistics interface IFNAME
Parameters	**IFNAME**: Interface name
Example usage	`switch_a# show dot1x interface ge5`

802.1x Configuration

dot1x initialize	
Purpose	Unauthorize a port, and attempt reauthentication on the specified interface.
Command Mode	Privileged Exec
Syntax	dot1x initialize interface IFNAME
Parameters	**IFNAME**: Interface name
Example usage	`switch_a# dot1x initialize interface ge5`

dot1x keytxenabled	
Purpose	Enable or disable key transmission over an Extensible Authentication Protocol (EAP) packet between the authenticator and supplicant.
Command Mode	Interface Configuration
Syntax	dot1x keytxenabled (enable | disable)
Parameters	None
Example usage	`switch_a(config-if)# dot1x keytxenabled disable`

dot1x port-control	
Purpose	Force a port state. Use the no parameter to remove a port from the 802.1x management.
Command Mode	Interface Configuration
Syntax	dot1x port-control dir (in | both) dot1x port-control (force-unauthorized | force-authorized | auto) no dot1x port-control
Parameters	**auto**: Enable authentication on port. **dir**: Specify the packet control direction. **both**: Discard receive and transmit packets from the supplicant **in**: Discard receive packets from the supplicant **force-authorized**: Force a port to always be in an authorized state. **force-unauthorized**: Force a port to always be in an unauthorized state.
Example usage	`switch_a(config-if)# dot1x port-control auto`

dot1x reauthentication	
Purpose	Enable reauthentication on a port.
Command Mode	Interface Configuration
Syntax	[no] dot1x reauthentication
Parameters	None
Example usage	`switch_a(config-if)# dot1x reauthentication`

dot1x system-auth-ctrl	
Purpose	Enable authentication globally.
Command Mode	Interface Configuration
Syntax	[no] dot1x system-auth-ctrl
Parameters	None
Example usage	`switch_a(config-if)# dot1x system-auth-ctrl`

dot1x protocol-version	
Purpose	Set the protocol version of dot1x to 1 or 2. The protocol version must be synchronized with the Xsupplicant being used in that interface. Use the no parameter to set the protocol version to the default value (2).
Command Mode	Interface Configuration
Syntax	dot1x protocol-version <1-2> no dot1x protocol-version
Parameters	**<1-2>**: EAP Over LAN (EAPOL) version.
Example usage	`switch_a(config-if)# dot1x protocol-version 2`

dot1x quiet-period	
Purpose	Set the quiet-period time interval. When a switch cannot authenticate a client, the switch remains idle for a quiet-period interval of time, then tries again. By administratively changing the quiet-period interval, by entering a lower number than the default, a faster response time can be provided. Use the no parameter to set the configured quiet period to the default (60 seconds).
Command Mode	Interface Configuration
Syntax	dot1x quiet-period <1-65535> no dot1x quiet-period
Parameters	**<1-65535>**: Seconds between the retrial of authentication.
Example usage	`switch_a(config-if)# dot1x quiet-period 200`

dot1x reauthMax	
Purpose	Set the maximum reauthentication value, which sets the maximum number of reauthentication attempts after which the port will be unauthorized. Use the no parameter to set the reauthentication maximum to the default value (2).
Command Mode	Interface Configuration
Syntax	dot1x reauthMax <1-10> no dot1x reauthMax
Parameters	**<1-10>**: Maximum number of reauthentication attempts after which the port will be unauthorized.
Example usage	`switch_a(config-if)# dot1x reauthMax 5`

dot1x timeout re-authperiod	
Purpose	Set the interval between reauthorization attempts. Use the no parameter to disable the interval between reauthorization attempts.
Command Mode	Interface Configuration
Syntax	dot1x timeout re-authperiod <1-4294967295> no dot1x timeout re-authperiod
Parameters	**<1-4294967295>**: Seconds between reauthorization attempts
Example usage	`switch_a(config-if)# dot1x timeout re-authperiod 25`

dot1x timeout server-timeout	
Purpose	Set the authentication sever response timeout.
Command Mode	Interface Configuration
Syntax	dot1x timeout server-timeout <1-65535> no dot1x timeout server-timeout
Parameters	**<1-65535>**: Authentication server response timeout
Example usage	`switch_a(config-if)# dot1x timeout server-timeout 555`

dot1x timeout supp-timeout	
Purpose	Set the interval for a supplicant to respond.
Command Mode	Interface Configuration
Syntax	dot1x timeout supp-timeout <1-65535> no dot1x timeout supp-timeout
Parameters	**<1-65535>**: Authentication supplicant response timeout
Example usage	`switch_a(config-if)# dot1x timeout supp-timeout 40`

dot1x timeout tx-period	
Purpose	Set the interval between successive attempts to request an ID.
Command Mode	Interface Configuration
Syntax	dot1x timeout tx-period <1-65535> no dot1x timeout tx-period
Parameters	**<1-65535>**: Interval between attempts.
Example usage	`switch_a(config-if)# dot1x timeout tx-period 34`

ip radius source-interface	
Purpose	Set the local address sent in packets to the radius server.
Command Mode	Global Configuration
Syntax	ip radius source-interface HOSTNAME PORT no ip radius source-interface
Parameters	**HOSTNAME**: Radius client in IP address or hostname format. **PORT**: Radius client port number. The default port number is 1812.
Example usage	`switch_a(config)# ip radius source-interface groohost 1812`

radius-server deadtime	
Purpose	Specify the time that a nonresponding radius server is passed over by requests for authentication. The no form sets the default (0).
Command Mode	Global Configuration
Syntax	radius-server deadtime MIN no radius-server deadtime
Parameters	**MIN**: Length of time (in minutes), maximum of 1440.
Example usage	`switch_a(config)# radius-server deadtime 10`

radius-server host	
Purpose	Specify the IP address or host name of the remote radius server host and assign authentication and accounting destination port numbers. Multiple radius-server host commands can be used to specify multiple hosts. The software searches for hosts in the order they are specified. If no host-specific timeout, retransmit, or key values are specified, the global values apply to that host. If the auth-port parameter is not specified, it will take the default value of the auth-port. If you do not specify the authport to unconfigure, and the default value of the auth-port does not match the port you are trying to unconfigure, the specified radius-server host will not be unconfigured.
Command Mode	Global Configuration
Syntax	radius-server host HOSTNAME radius-server host HOSTNAME {key STRING | retransmit RETRIES | timeout SEC | auth-port PORTNO} no radius-server host HOSTNAME (auth-port PORT|)
Parameters	**auth-port**: (Optional) Specify the UDP destination port for authentication requests; the host is not used for authentication if set to 0. **key**: (Optional) Specify the authentication and encryption key for all radius communications between the router and the radius server. This key must match the encryption used on the radius daemon. All leading spaces are ignored, but spaces within and at the end of the string are used. If spaces are used in the string, do not enclose the string in quotation marks unless the quotation marks themselves are part of the key. **retransmit**: (Optional) The number of times a radius request is re-sent to a server, if that server is not responding or responding slowly. This setting overrides the global setting of the radiusserver retransmit command. Enter a value in the range 1 to 100. If no retransmit value is specified, the global value is used. **timeout**: (Optional) The time interval (in seconds) that the router waits for the radius server to reply before retransmitting. This setting overrides the global value of the radius-server timeout command. If no timeout value is specified, the global value is used. Enter a value in the range 1 to 1000.
Example usage	`switch_a(config)# radius-server host 10.10.10.40 auth-port 1812 timeout 5 retransmit 3 key authd`

radius-server key	
Purpose	Set the shared secret key between a Radius server and a client.
Command Mode	Global Configuration
Syntax	radius-server key KEY no radius-server key
Parameters	**KEY**: The secret key shared among the radius server and the 802.1x client.
Example usage	`switch_a(config)# radius-server key ipi`

radius-server retransmit	
Purpose	Specify the number of times the router transmits each radius request to the server before giving up.
Command Mode	Global Configuration
Syntax	radius-server retransmit RETRIES no radius-server retransmit
Parameters	**RETRIES**: The retransmit value from 1 to 100. If no retransmit value is specified, the global value is used.
Example usage	`switch_a(config)# radius-server retransmit 12`

radius-server timeout	
Purpose	Specify the number of seconds a router waits for a reply to a radius request before retransmitting the request. Use the no parameter to use the default value.
Command Mode	Global Configuration
Syntax	radius-server timeout SEC no radius-server timeout
Parameters	**SEC**: The number of seconds for a router to wait for a server host to reply before timing out. <1-1000>
Example usage	`switch_a(config)# radius-server timeout 20`

14 LLDP (Link Layer Discovery Protocol)

LLDP Information

show lldp	
Purpose	Show statistics, status, and information for current LLDP configuration.
Command Mode	Privileged exec
Syntax	show lldp entry WORD show lldp statistics show lldp statistics IFNAME show lldp interface show lldp interface IFNAME show lldp neighbors show lldp neighbors-org show lldp neighbors IFNAME show lldp neighbors detailed
Parameters	**IFNAME**: Interface name **WORD**: System Name of LLDP neighbor entry
Example usage	`switch_a# show lldp statistics`

LLDP Configuration

lldp enable	
Purpose	Enable lldp.
Command Mode	Global Configuration
Syntax	[no] lldp enable
Parameters	None
Example usage	`switch_a(config)# lldp enable`

lldp holdtime multiplier	
Purpose	Set the holdtime multiplier value, which is multiplied by the transmit interval to calc Time To Live (TTL) that is advertised to neighbors
Command Mode	Global Configuration
Syntax	lldp holdtime multiplier <2-10> no lldp holdtime multiplier
Parameters	**<2-10>**: Multiplier factor
Example usage	`switch_a(config)# lldp holdtime multiplier 5`

lldp txinterval	
Purpose	Set interval at which LLDP frames are transmitted.
Command Mode	Global Configuration
Syntax	lldp txinterval <5-32768> no lldp txinterval
Parameters	**<5-32768>**: TxInterval, default is 30 sec
Example usage	`switch_a(config)# lldp txinterval 100`

lldp tlv-global	
Purpose	Configure the global TLV (Time – Length – Value) settings are advertised by the switch to other LLDP devices.
Command Mode	Global Configuration
Syntax	lldp tlv-global {all | port-descr | sys-name | sys-descr | sys-cap | mgmt-addrs | port-vlan-id | mac-phy | protocol-identity | vlan-name | port-and-protocol | link-aggregation | max-frame}
Parameters	**port-descr**: Port Description **sys-name**: System Name TLV **sys-descr** : System Description TLV **sys-cap**: System Capabilities **mgmt-addrs**: Management Address **port-vlan-id**: Port VLAN ID **mac-phy**: MAC/PHY Configuration/Status **port-and-protocol**: Port And Protocol VLAN ID **vlan-name**: VLAN Name **protocol-identity**: Protocol Identity **link-aggregation**: Link Aggregation **max-frame**: Maximum Frame Size
Example usage	`switch_a(config)# lldp tlv-global sys-name`

LLDP Port Settings

lldp tx-pkt	
Purpose	Enable LLDP transmit on a port.
Command Mode	Interface Configuration
Syntax	[no] lldp tx-pkt
Parameters	None
Example usage	`switch_a(config-if)# lldp tx-pkt`

lldp tx-rcv	
Purpose	Enable LLDP receive on a port.
Command Mode	Interface Configuration
Syntax	[no] lldp tx-rcv
Parameters	None
Example usage	`switch_a(config-if)# lldp tx-rcv`

lldp mgmt-ip vlan	
Purpose	Enable the transmission of the management IP address through a port.
Command Mode	Interface Configuration
Syntax	lldp mgmt-ip vlan <1-4094> no lldp mgmt-ip vlan
Parameters	**<1-4094>**: VLAN ID
Example usage	`switch_a(config-if)# lldp mgmt-ip vlan 200`

lldp notification	
Purpose	Enable LLDP notification
Command Mode	Interface Configuration
Syntax	[no] lldp notification
Parameters	None
Example usage	`switch_a(config-if)# lldp notification`

lldp-agent	
Purpose	Configure lldp agents for customer-bridge and non-TPMR-bridge.
Command Mode	Interface Configuration
Syntax	[no] lldp-agent customer-bridge [no] lldp-agent non-tpmr-bridge
Parameters	None
Example usage	`switch_a(config-if)# lldp-agent customer-bridge`

lldp tlv-global	
Purpose	Configure the global TLV (Time – Length – Value) settings are advertised by the switch to other LLDP devices.
Command Mode	Interface Configuration
Syntax	lldp tlv-global {all|port-descr | sys-name | sys-descr | sys-cap | mgmt-addrs | port-vlan-id | mac-phy | protocol-identity | vlan-name | port-and-protocol | link-aggregation | max-frame}
Parameters	**port-descr**: Port Description **sys-name**: System Name TLV **sys-descr** : System Description TLV **sys-cap**: System Capabilities **mgmt-addrs**: Management Address **port-vlan-id**: Port VLAN ID **mac-phy**: MAC/PHY Configuration/Status **port-and-protocol**: Port And Protocol VLAN ID **vlan-name**: VLAN Name **protocol-identity**: Protocol Identity **link-aggregation**: Link Aggregation **max-frame**: Maximum Frame Size
Example usage	`switch_a(config-if)# lldp tlv-global sys-name`

15 DHCP (Dynamic Host Configuration Protocol)

DHCP Server

show dhcp-client status	
Purpose	Display DHCP-client Current Status
Command Mode	Privileged exec
Syntax	show dhcp-client status
Parameters	None
Example usage	`switch_a# show dhcp-client status`

show running-config dhcp	
Purpose	Display DHCP-client Current Status
Command Mode	Global configuration
Syntax	show running-config dhcp
Parameters	None
Example usage	`switch_a(config)# show running-config dhcp`

feature dhcp	
Purpose	Enable/disable dhcp on switch
Command Mode	Global configuration
Syntax	[no] feature dhcp
Parameters	none
Example usage	`switch_a(config)#feature dhcp`

ip address dhcp	
Purpose	Get an IP address from a DHCP server for this interface. Use the no form to disable the DHCP client for this interface.
Command Mode	Interface configuration
Syntax	[no] ip address dhcp
Parameters	none
Example usage	`switch_a(config-if)# ip address dhcp`

ip dhcp client request	
Purpose	Request a DNS nameserver or host name for DHCP client.
Command Mode	Interface configuration
Syntax	ip dhcp client request dns-nameserver no ip dhcp client request dns-nameserver ip dhcp client request host-name no ip dhcp client request host-name
Parameters	**dns-nameserver**: List of DNS name servers **host-name**: Name of the client
Example usage	`switch_a(config-if)# ip dhcp client request dns-nameserver`

16 NTP (Network Time Protocol)

NTP Configuration

show system time	
Purpose	Display system time
Command Mode	Privileged Exec
Syntax	show system time
Parameters	none
Example usage	`switch_a# show system time`

show ntp associations	
Purpose	Display NTP associations
Command Mode	Privileged Exec
Syntax	show ntp associations
Parameters	none
Example usage	`switch_a# show ntp associations`

show ntp status	
Purpose	Display ntp status.
Command Mode	Privileged Exec
Syntax	show ntp status
Parameters	none
Example usage	`switch_a# show ntp status`

set clock	
Purpose	Set time and date for the switch.
Command Mode	Privileged Exec
Syntax	set clock <2000-2037> <1-12> <1-31> <0-23> <0-59> <0-59>
Parameters	**<2000-2037>**: Year, **<1-12>**: Month, **<1-31>**: Date, **<0-23>**: Hour **<0-59>**: Minute , **<0-59>**: Second
Example usage	`switch_a# set clock 2017 3 27 17 24 30`

ntp enable	
Purpose	Enable ntp
Command Mode	Global Configuration
Syntax	[no] ntp enable
Parameters	None
Example usage	`switch_a(config)# ntp enable`

ntp sync-time	
Purpose	Have the NTP client sync the clock immediately switch,
Command Mode	Global Configuration
Syntax	ntp sync-time
Parameters	None
Example usage	`switch_a(config)# ntp sync-time`

clock	
Purpose	Set time zone for the switch
Command Mode	Global Configuration
Syntax	clock timezone <ZONE> <OFFSET>
Parameters	**ZONE**: Name of time zone (Examples: CST, MST, PST, UCT, EST, MST, RFT, IST, NAST, TTST, MET, NZST, UAEST)pst **OFFSET**: Offset from Coordinated Universal Time (UTC), range is <0~24>:<0~59>
Example usage	`switch_a(config)# clock timezone pst 9:0`

clock summer time	
Purpose	Set time zone, system daylight saving day, and system daylight saving weekday mode
Command Mode	Global Configuration
Syntax	clock summer-time ZONE { date { date month year hh:mm date month year hh:mm | month date year hh:mm month date year hh:mm } | recurring week day month hh:mm week day month hh:mm } [1-480] no clock summer-time
Parameters	**ZONE**: Name of the time zone (for example, PDT) displayed when summer time is in effect. **date**: Indicates that summer time should start on the first specific date listed in the command and end on the second specific date in the command. **date**: Date of the month. **recurring**: Indicates that summer time should start and end on the corresponding specified days every year. **week**: Week of the month <1-5> **[1-480]**: Number of minutes to add during summer time.
Example usage	`switch_a(config)# clock summer-time CDT weekday 2 Sun March 2 0 1 Sun November 2 0 60`

ntp server	
Purpose	Configure NTP server, server peer authentication key, peer preference, and NTP server version.
Command Mode	Global Configuration
Syntax	ntp server <WORD> ntp server <WORD> key <1-4294967295> ntp server <WORD> prefer ntp server <WORD> version <1-4> no ntp server <WORD>
Parameters	**WORD**: IP address or host name of server
Example usage	`switch_a(config)# ntp server 10.10.10.10`

Static Route

show ip route	
Purpose	Display the IP routing table for a protocol or from a particular table.
Command Mode	Privileged Exec
Syntax	show ip route show ip route <A.B.C.D> show ip route <A.B.C.D/M> show ip route database show ip route static show ip route summary
Parameters	**A.B.C.D**: Display network in the IP routing table. **A.B.C.D/M**: Display IP prefix <network>/<length>, for example, 35.0.0.0/8. **database**: Display IPv6 routing table database information. **static**: Display static routes. **summary**: Display a summary of all routes
Example usage	`switch_a# show ip route 10.10.10.5`

show routing	
Purpose	Display routing information,
Command Mode	Privileged Exec
Syntax	show routing show routing <A.B.C.D> show routing <A.B.C.D/M> show routing database show routing static show routing summary
Parameters	**A.B.C.D**: Display network in the IP routing table. **A.B.C.D/M**: Display IP prefix <network>/<length>, Example - 35.0.0.0/8. **database**: Display IPv6 routing table database information. **static**: Display static routes. **summary**: Display a summary of all routes
Example usage	`switch_a# show routing 10.10.10.10/24`

ip route	
Purpose	Create an IPv4 static route. Use the no form to delete a static route.
Command Mode	Global Configuration
Syntax	[no] ip route <A.B.C.D/M> <A.B.C.D> [no] ip route <A.B.C.D> <A.B.C.D> <A.B.C.D> [no] ip route <A.B.C.D/M> <A.B.C.D> <1-255> [no] ip route <A.B.C.D/M> <A.B.C.D> description <WORD> [no] ip route <A.B.C.D/M> <A.B.C.D> tag <1-4294967295> [no] ip route <A.B.C.D/M> <IFNAME> [A.B.C.D |]
Parameters	**A.B.C.D/M**: Subnet IP destination prefix and mask <0-32> **A.B.C.D A.B.C.D**: Subnet: IP destination address and mask **A.B.C.D**: Gateway nexthop IPv4 address **<1-255>**: Administrative distance **IFNAME**: Gateway nexthop interface name **description**: Description of the static route **tag**: Tag used as a "match" value to control redistribution via route maps **<1-4294967295>**: Tag value
Example usage	`switch_a(config)# ip route 192.168.3.0 255.255.255.0 2.2.2.2 128`

ip static	
Purpose	Enable BFD support on a static route. Use the no form to delete a static route.
Command Mode	Global Configuration
Syntax	[no] ip static [A.B.C.D/M | A.B.C.D | fall-over | bfd]
Parameters	**A.B.C.D/M**: IP destination prefix and mask <0-32> **A.B.C.D**: IP gateway address **fall-over**: Specify fall-over detection **bfd**: Specify Bidirectional Forwarding Detection (BFD)
Example usage	`switch_a(config)# ip static 4.4.4.4/32 20.0.10.82 fall-over bfd`

max-static-routes	
Purpose	Set maximum static routes number.
Command Mode	Global Configuration
Syntax	max-static-routes <1-8192>
Parameters	**<1-8192>**: Allowed number of static routes
Example usage	`switch_a(config)# max-static-routes 100`

ip prefix-list	
Purpose	Create a prefix list. Prefixes are matched from the top of the prefix list, and matching stops whenever a match or deny occurs. For efficiency, use the **seq** parameter and place common matches or denials towards the top of the list. The sequence values are generated in the sequence of 5. The parameters GE and LE specify the range of the prefix length to be matched. When setting these parameters, set LE to be less than 32 and GE to be less than LE.
Command Mode	Global Configuration
Syntax	[no] ip prefix-list WORD (deny | permit) (A.B.C.D/M | any) [no] ip prefix-list WORD (deny | permit) A.B.C.D/M ge <0-32> [no] ip prefix-list WORD (deny | permit) A.B.C.D/M ge <0-32> le <0-32> [no] ip prefix-list WORD (deny | permit) A.B.C.D/M le <0-32> [no] ip prefix-list WORD (deny | permit) A.B.C.D/M le <0-32> ge <0-32> [no] ip prefix-list WORD seq <1-4294967295> (deny | permit) (A.B.C.D/M | any) [no] ip prefix-list WORD seq <1-4294967295> (deny | permit) A.B.C.D/M ge <0-32> [no] ip prefix-list WORD seq <1-4294967295> (deny | permit) A.B.C.D/M ge <0-32> le <0-32> [no] ip prefix-list WORD seq <1-4294967295> (deny | permit) A.B.C.D/M le <0-32> [no] ip prefix-list WORD seq <1-4294967295> (deny | permit) A.B.C.D/M le <0-32> ge <0-32> [no] ip prefix-list sequence-number [no] ip prefix-list WORD description LINE
Parameters	**WORD:** Specify the name of a prefix list. **deny:** Specify that packets are to be rejected. **Description**: Prefix-list specific description. **LINE:** Up to 80 characters describing this prefix-list **permit:** Specify that packets are to be accepted. **A.B.C.D/M:** The IP address mask and length of the prefix list mask **le:** Maximum prefix length to be matched <0-32>. **ge:** Minimum prefix length to be matched <0-32>. **seq:** The sequence number of the prefix list <1-429496725>. **any:** Takes all packets of any length. **sequence-number:** Sequence numbers in nonvolatile generation
Example usage	`switch_a(config)# ip prefix-list mylist seq 5 deny 76.2.2.0/24`

Route Table

show route-table	
Purpose	Display the route table, which contains information about the topology of the surrounding network.
Command Mode	Privileged Exec
Syntax	show route-table
Parameters	None
Example usage	`switch_a# show route-table`

Route Map

show route-map	
Purpose	Display the IP routing table for a protocol or from a particular table.
Command Mode	Privileged Exec
Syntax	show route map [WORD]
Parameters	**WORD:** Route map name
Example usage	`switch_a# show route map`

route-map	
Purpose	Enter route-map mode, and configure permit or deny match/set operations. This command controls and modifies routing information to allow redistribution of routes. It has a list of **match** and **set** commands associated with it. The match commands specify the conditions under which redistribution is allowed, and the set commands specify the redistribution actions to be performed if the match criteria are met. Route allow for detailed control over route distribution between routing processes. Route maps also allow policy routing, and can route packets to a different route than the obvious shortest path.
Command Mode	Global configuration
Syntax	route-map WORD (deny | permit) <1-65535> no route-map WORD ((deny | permit) <1-65535>|)
Parameters	**WORD**: Identify the route. **deny**: Route map denies set operations. If the deny parameter is specified, and the match criteria are met, the route is not redistributed, and any other route maps with the same map tag are not examined. **permit**: Route map permits set operations. If the permit parameter is specified, and the match criteria are met, the route is redistributed as specified by set actions. If the match criteria are not met, the next route map with the same tag is tested. **<1-65535>**: Sequence to insert to or delete from an existing route-map entry
Example usage	`switch_a(config)# route-map permit 100` `switch_a(config-route-map)#`

match interface	
Purpose	Define interface match criterion. Specifies the next-hop interface name of a route to be matched.
Command Mode	Route map
Syntax	match interface IFNAME no match interface (IFNAME|)
Parameters	**IFNAME**: Interface name.

Example usage	`switch_a#configure terminal`
	`switch_a(config)#route-map exampleroute permit 10`
	`switch_a(config-route-map)#match interface ge5`

match ip	
Purpose	Match address of a route, a prefix-list, a next-hop address, the next-hop IP address using the prefix-list, a peer IPv4 address of a route.
Command Mode	Route map
Syntax	[no] match ip address (<1-199> | <1300-2699> | WORD) [no] match ip address prefix-list (WORD) [no] match ip next-hop (<1-199> | <1300-2699> | WORD) [no] match ip next-hop prefix-list (WORD) [no] match ip peer (<1-199> | <1300-2699> | WORD)
Parameters	**WORD: IP access-list name** **<1-199>: IP access-list number (standard range)** **<1300-2699>: IP access-list number (expanded range)**

Example usage	`switch_a#configure terminal`
	`switch_a(config)# route-map rmap1 permit 3`
	`switch_a(config-route-map)# match ip address prefix-list mylist`

match metric	
Purpose	Match a metric of a route. The route specified by the policies might not be the same as specified by the routing protocols. Setting policies enable packets to take different routes depending on their length or content. Packet forwarding based on configured policies overrides packet forwarding specified in routing tables.
Command Mode	Route map
Syntax	match metric <0-4294967295> no match metric (<0-4294967295> |)
Parameters	**<0-4261412864>: Metric value**

Example usage	`switch_a#configure terminal`
	`switch_a(config)# route-map mapexample permit 3`
	`switch_a(config-route-map)# match metric 223455`

Note	This command is valid for BGP, OSPF, RIP, and IS-IS only.

set ip next-hop	
Purpose	Set the specified next-hop value.
Command Mode	Route map
Syntax	set ip next-hop A.B.C.D (interface IFNAME |) (primary | secondary |) no set ip next-hop (A.B.C.D |) (interface IFNAME |) (primary | secondary |)
Parameters	**A.B.C.D**: IP address of the next-hop. **Interface**: Interface name **primary**: Specify the nexthop as primary. **secondary**: Specify the nexthop as secondary.
Example usage	`switch_a(config)# route-map rmap1 permit 3` `switch_a(config-route-map)# set next-hop 10.10.0.67`
Note	This command is valid for BGP, OSPF, and RIP only.

set metric	
Purpose	Set a metric value for a route and influence external neighbors about the preferred path into an Autonomous System (AS). The preferred path is the one with a lower metric value. A router compares metrics for paths from neighbors in the same ASs. To use this command, you must first have a match clause. **Match** and **set** commands set conditions for redistributing routes from one routing protocol to another.
Command Mode	Route map
Syntax	[no] set metric (<0-4294967295>|<+/-metric>|)
Parameters	**<0-4294967295>**: Specify a metric value. **<+/-metric>**: Adds or subtracts a metric.
Example usage	`switch_a(config)# route-map rmap1 permit 3` `switch_a(config-route-map)# set metric 600`

Proxy ARP

ip proxy-arp	
Purpose	Enable proxy ARP on an interface, allowing the switch to answer ARP queries for a network address that is not on that network.
Command Mode	Interface Configuration
Syntax	[no] ip proxy-arp
Parameters	None
Example usage	`switch_a(config-if)# ip proxy arp`

18 RIP (Routing Information Protocol)

RIP Information and General Settings

show ip rip	
Purpose	Display RIP configuration info.
Command Mode	Privileged Exec
Syntax	show ip rip show ip rip database show ip rip interface show ip rip interface <IFNAME> show ip rip statistics show ip rip statistics <IFNAME>
Parameters	**IFNAME**: Interface name **database**: IP RIP database
Example usage	`switch_a# show ip rip interface`

show ip protocols rip	
Purpose	Show the RIP process parameter and statistics information.
Command Mode	Privileged Exec
Syntax	show ip protocols rip
Parameters	None
Example usage	`switch_a# show ip protocols rip`

clear ip rip route	
Purpose	Clear specific data from the RIP routing tables. Using this command with the all parameter clears the RIP table of all routes. To prevent the RIP from being deleted, use the **redistribute connected** command and make the RIP network a connected route. To delete the RIP routes learned from neighbors and also keep the RIP network intact, use the **rip** parameter (**clear ip rip route rip**).
Command Mode	Privileged Exec
Syntax	clear ip rip route (A.B.C.D/M | rip | kernel | connected | static | ospf | isis | bgp | all)
Parameters	**A.B.C.D/M**: Removes entries which exactly match this destination address **bgp**: Removes BGP routes from the RIP routing table. connected: Removes entries for connected routes **isis**: Removes IS-IS routes **kernel**: Removes kernel entries. **ospf**: Removes only OSPF routes. **rip**: Removes only RIP routes. **static**: Removes static entries. **all**: Removes the entire RIP routing table.
Example usage	`switch_a# clear ip rip route 10.0.0.0/8`

clear ip rip statistics	
Purpose	Clear RIP statistics.
Command Mode	Privileged Exec
Syntax	clear ip rip statistics (A.B.C.D/M | rip | kernel | connected | static | ospf | isis | bgp | all)
Parameters	**A.B.C.D/M**: Removes entries which exactly match this destination address **bgp**: Removes BGP routes from the RIP routing table. connected: Removes entries for connected routes **isis**: Removes IS-IS routes **kernel**: Removes kernel entries. **ospf**: Removes only OSPF routes. **rip**: Removes only RIP routes. **static**: Removes static entries. **all**: Removes the entire RIP routing table.
Example usage	`switch_a# clear ip rip statistics ospf`

router rip	
Purpose	Enable RIP routing. This command places the switch into Router Configuration Mode.
Command Mode	Global Configuration
Syntax	[no] router rip
Parameters	None
Example usage	`switch_a(config)#router rip` `switch_a(config-router)#`

version	
Purpose	Specify a RIP version to be used globally. Options are version 1 or version 2. Version 2 has more features than version 1, including authentication. Once the rip version is set, rip packets of that version will be received and sent on all the rip-enabled interfaces. Use the no parameter to restore the default version (version 2).
Command Mode	Router Configuration
Syntax	version <1-2>
Parameters	**<1-2>**: RIP version
Example usage	`switch_a(config-router)#version 1`

default-information originate	
Purpose	Distribute a default route, always advertise default route, create a route map reference.
Command Mode	Router Configuration
Syntax	[no] default-information originate default-information originate always default-information originate route-map <WORD> default-information originate always route-map <WORD>
Parameters	**always**: Always advertise default route **route-map**: Route map reference **WORD**: Pointer to route-map entries
Example usage	`switch_a(config-router)# default-information originate always`

bfd all-interfaces	
Purpose	Enable BFD on all interfaces associated with the routing process.
Command Mode	Router Configuration
Syntax	bfd all-interfaces
Parameters	None
Example usage	`switch_a(config-router)# bfd all-interfaces`

cisco-metric-behavior	
Purpose	Enable metric updates consistent with Cisco.
Command Mode	Router Configuration
Syntax	cisco-metric-behavior <enable/disable>
Parameters	None
Example usage	`switch_a(config-router)# cisco-metric-behavior enable`

default-metric	
Purpose	Specify the metrics to be assigned to redistributed routers. This command is used in conjunction with the **redistribute** command to make the routing protocol use the specified metric value for all redistributed routes. A default metric is useful in redistributing routes with incompatible metrics. Every protocol has different metrics and can not be compared directly. Default metric provides the standard to compare. All routes that are redistributed will use the default metric.
Command Mode	Router Configuration
Syntax	default-metric <1-16>
Parameters	**<1-16>**: default metric.
Example usage	`switch_a(config-router)# default-metric 10`

distribute-list	
Purpose	Filter incoming or outgoing route updates using an access list or a prefix list. Incoming or outgoing route updates can be filtered out using an access list or a prefix list. If no interface is specified, the filter will be applied to all the interfaces.
Command Mode	Router Configuration
Syntax	distribute-list WORD (in | out) (IFNAME |) distribute-list prefix WORD (in | out) (IFNAME |) no distribute-list WORD (in | out) (IFNAME |) no distribute-list prefix WORD (in | out) (IFNAME |)
Parameters	**WORD**: The IPv4 access-list number or name to use. **prefix**: Filter prefixes in routing updates. **WORD**: The name of the IPv4 prefix-list to use. **in**: Filter incoming routing updates. **out**: Filter outgoing routing updates. **IFNAME**: The name of the interface on which distribute-list applies.
Example usage	`switch_a(config-router)# distribute-list prefix myfilter in ge10`

maximum-prefix	
Purpose	Set the maximum number of RIP routes, and the percentage of maximum routes created that will generate a warning (Default 75%).
Command Mode	Router Configuration
Syntax	maximum-prefix <1-65535> maximum-prefix <1-65535> (1-100) no maximum-prefix
Parameters	**<1-65535>**: The maximum number of RIP routes allowed. **<1-100>**: Percentage of maximum routes that will generate a warning. Default is 75%.
Example usage	`switch_a(config-router)# maximum-prefix 150`

timers basic	
Purpose	Adjusts the RIP timing parameters. Every 30 seconds, an update is sent out containing the complete routing table to every neighboring router. When the time specified by the timeout parameter expires, the route is no longer valid. However, it is retained in the routing table for a short time so that neighbors are notified that the route has been dropped. When the time specified by the garbage parameter expires, the route is finally removed from the routing table. Until the garbage time expires, the route is included in all updates sent by the router. All routers in the network must have the same timers to allow RIP to execute distributed and asynchronous routing algorithms. The timers should not be synchronized as it might lead to unnecessary collisions on the network.
Command Mode	Router Configuration
Syntax	timers basic: <5-2147483647> <5-2147483647> <5-2147483647> no timers basic
Parameters	**<5-2147483647>**: The routing table update timer in seconds. Default is 30. **<5-2147483647>**: The routing information timeout timer in seconds. Default is 180 seconds. After this interval has elapsed and no updates for a route are received, the route is declared invalid. **<5-2147483647>**: The routing garbage collection timer in seconds. Default is 120 seconds.
Example usage	`switch_a(config-router)# timers basic 30 180 120`

recv-buffer-size	
Purpose	Configure the RIP UDP receive-buffer size.
Command Mode	Router Configuration
Syntax	recv-buffer-size <8192-2147483647> no recv-buffer-size (<8192-2147483647> |)
Parameters	**<8192-2147483647>**: The RIP UDP receive buffer size value
Example usage	`switch_a(config-router)# recv-buffer-size 150000`

RIP Interface Settings

ip rip authentication key-chain	
Purpose	Enable RIP version 2 authentication on an interface and specify the name of the key chain to be used. If no key chain is configured, then the result will be no authentication.
Command Mode	Interface Configuration
Syntax	ip rip authentication key-chain LINE no ip rip authentication key-chain
Parameters	**LINE**: The name of the key chain.
Example usage	`switch_a(config-if)# ip rip authentication key-chain rufferto`

ip rip authentication mode	
Purpose	Specify the type of authentication mode used for RIP version 2 packets
Command Mode	Interface Configuration
Syntax	ip rip authentication mode md5 ip rip authentication mode text no ip rip authentication mode
Parameters	**md5**: Uses the keyed MD5 authentication algorithm. **text**: The clear text or simple password authentication.
Example usage	`switch_a(config-if)# ip rip authentication mode md5`

ip rip authentication string	
Purpose	Specify the authentication string or password used by a key. You can choose to configuring authentication for single key or multiple keys at different times. Use this command to specify password for a single key on an interface.
Command Mode	Interface Configuration
Syntax	ip rip authentication string LINE no ip rip authentication string
Parameters	**LINE**: The authentication string or password used by a key.
Example usage	`switch_a(config-if)# ip rip authentication string guest`

ip rip receive version	
Purpose	Receive specified version of RIP packets on an interface basis using version control, and override the setting of the version command. Use no form to set default (2).
Command Mode	Interface Configuration
Syntax	ip rip receive version (1|2) ip rip receive version 1 2 ip rip receive version 2 1 no ip rip receive version
Parameters	**1**: Accept RIP version 1 packets on the interface. **2**: Accept RIP version 2 packets on the interface. **1 2**: Accept RIP version 1 and 2 packets on the interface. **2 1**: Accept RIP version 2 and 1 packets on the interface.
Example usage	`switch_a(config-if)# ip rip receive version 1 2`

ip rip receive-packet	
Purpose	Configure the interface to enable the reception of RIP packets.
Command Mode	Interface Configuration
Syntax	[no] ip rip receive-packet
Parameters	None
Example usage	`switch_a(config-if)# ip rip receive-packet`

ip rip send version	
Purpose	Interface version control. In addition to version 1 & 2, compatible version packets can be specified. The parameter **1-compatible** lets a version 2 RIP interface broadcast packets instead of multicasting. This command overrides version specified by the **version** command. Use no parameter for global RIP version rules.
Command Mode	Interface Configuration
Syntax	ip rip send version (1 | 2 |1-compatible) ip rip send version 1 2 ip rip send version 2 1 no ip rip send version
Parameters	1: Send RIP version 1 packets out of an interface. 2: Send RIP version 2 packets out of an interface. 1-compatible: Send RIP version 1 compatible packets from a version 2 RIP interface.

Example usage	`switch_a(config-if)# ip rip send version 1 2`
ip rip send-packet	
Purpose	Enable the sending of RIP packets through the current interface.
Command Mode	Interface Configuration
Syntax	[no] ip rip send-packet
Parameters	None
Example usage	`switch_a(config-if)# ip rip send-packet`

ip rip split-horizon	
Purpose	Perform the split-horizon action on the interface. This command helps avoid including routes in updates sent to the same gateway from which they were learned. Using the split horizon command omits routes learned from one neighbor, in updates sent to that neighbor. Using the **poisoned** parameter includes such routes in updates, but sets their metrics to infinity (effectively advertising that these routes are unreachable).
Command Mode	Interface Configuration
Syntax	ip rip split-horizon ip rip split-horizon poisoned no ip rip split-horizon
Parameters	**poisoned**: Performs split-horizon with poisoned reverse.
Example usage	`switch_a(config-if)# ip rip split-horizon poisoned`

RIP Route

offset-list	
Purpose	Add an offset to in and out metrics to routes learned through RIP. This command specifies the offset value that is added to the routing metric. When the networks match the access list the offset is applied to the metrics. No change occurs if the offset value is zero.
Command Mode	Router Configuration
Syntax	offset-list WORD (in | out) <0-16> (IFNAME|) no offset-list WORD (in | out) <0-16> (IFNAME|)
Parameters	**WORD**: The access-list number or names to apply. **in**: Access list will be used for metrics of incoming advertised routes. **out**: Access list will be used for metrics of outgoing advertised routes. **<0-16>**: Offset used for metrics of networks matching the access list. **IFNAME**: The interface to match.
Example usage	Set the router to examine the RIP updates being sent out from interface ge1 and add 16 hops to routes matching the ip addresses specified in the access list **accesslist1**.
	`switch_a(config-router)# offset-list accesslist1 in 16 ge1`

route	
Purpose	Configure static RIP routes. This command is used most often for debugging purposes and does not show up in the kernel routing table. After adding the RIP route, it can be checked in the RIP routing table.
Command Mode	Router Configuration
Syntax	[no] route A.B.C.D/M
Parameters	**A.B.C.D/M**: The IP address prefix and length.
Example usage	`switch_a(config-router)# route 10.10.10.0/24`

RIP Network

network	
Purpose	Specify a network as one that runs RIP. This command specifies the networks to which routing updates will be sent and received. If a network is not specified, the interfaces in that network will not be advertised in any RIP update.
Command Mode	Router Configuration
Syntax	[no] network A.B.C.D/M [no] network IFNAME
Parameters	**A.B.C.D/M**: The IP address prefix and length of this IP network. **IFNAME**: Alphanumeric string that defines the interface name.
Example usage	`switch_a(config-router)# network 10.0.0.0/8`

RIP Neighbor

neighbor	
Purpose	Specify a neighbor router. It is used for each connected point-to-point link. This command to exchanges non-broadcast routing information. It can be used multiple times for additional neighbors. The **passive-interface** command disables sending routing updates on an interface. Use the neighbor command in conjunction with the passive-interface command to send routing updates to specific neighbors.
Command Mode	Router Configuration
Syntax	neighbor <A.B.C.D> neighbor <A.B.C.D> fall-over bfd no neighbor A.B.C.D
Parameters	**A.B.C.D**: IP address of a neighboring router with which the routing information will be exchanged. **fall-over**: Fall-over detection **bfd**: Bidirectional Forwarding Detection
Example usage	`switch_a(config-router)# neighbor 20.20.20.20 fall-over bfd`

RIP Passive

passive-interface	
Purpose	Block RIP broadcast on the interface.
Command Mode	Router Configuration
Syntax	passive-interface IFNAME no passive-interface IFNAME
Parameters	**IFNAME**: Interface name.
Example usage	`switch_a(config-router)# passive-interface ge5`

RIP Redistribute

redistribute	
Purpose	Redistribute information from other routing protocols
Command Mode	Router Configuration
Syntax	[no] redistribute (kernel | connected | static | ospf | isis | bgp) [no] redistribute (kernel | connected | static | ospf | isis | bgp) metric <0-16> [no] redistribute (kernel | connected | static | ospf | isis | bgp) route-map WORD [no] redistribute (kernel | connected | static | ospf | isis | bgp) metric <0-16> route-map WORD
Parameters	**bgp**: Redistribute from BGP routes **connected**: Redistribute from connected routes **isis**: Redistribute from ISO IS-IS routes **kernel**: Redistribute from kernel routes **ospf6**: Redistribute from OSPF routes (version 3) **static**: Redistribute from static routes **metric**: Set metric value **<0–16>**: Metric value **route-map**: Route map reference **WORD**: Name of the route-map
Example usage	`switch_a(config-router)# redistribute static metric 8`

19 OSPF (Open Shortest Path First)

OSPF Information

show ip ospf	
Purpose	Display OSPF information, border and boundary router Information, details and summary of the OSPF database, interfaces, and multi-area adjacencies. View neighbor list, OSPF routing table, and virtual link information.
Command Mode	Privileged exec
Syntax	show ip ospf (<0-65535> |) show ip ospf (<0-65535> |) border-routers show ip ospf <0-65535> database(self-originate | max-age | adv-router A.B.C.D |) show ip ospf <0-65535> database (asbr-summary | external | network | router | summary | nssa-external | opaque-link | opaque-area | opaque-as) A.B.C.D (self-originate | adv-router A.B.C.D |) show ip ospf interface (IFNAME |) show ip ospf (<0-65535> |) multi-area-adjacencies show ip ospf (<0-65535> |) {neighbor | neighbor all | neighbor interface A.B.C.D | neighbor A.B.C.D | neighbor A.B.C.D detail | neighbor detail | neighbor detail all} show ip ospf (<0-65535> |) route (A.B.C.D | A.B.C.D/M | summary |) show ip ospf (<0-65535> |) virtual-links
Parameters	**<0-65535>**: The ID of the router process for which information will be displayed. **self-originate**: Self-originated link states. **max-age**: LSAs which have reached the max-age (3600 seconds). **adv-router**: Advertising router link states. **asbr-summary**: Autonomous System Boundary Router (ASBR) summary LSAs. **external**: External LSAs. **network**: Network LSAs. **router**: Router LSAs. **summary**: LSA summary information. **nssa-external**: NSSA external LSAs. **opaque-link**: Type 9 LSAs which are not flooded beyond the local network. **opaque-area**: Type 10 LSAs which are not flooded beyond the borders of their area. **opaque-as**: Type 11 LSAs which are flooded throughout the Autonomous System (AS).
Example usage	`switch_a# show ip ospf 500 database asbr-summary`

show ip protocols	
Purpose	Display OSPF process parameters and statistics.
Command Mode	Privileged exec
Syntax	show ip protocols [ospf]
Parameters	None
Example usage	`switch_a# show ip protocols ospf`

OSPF Configuration

router ospf	
Purpose	Enter router mode to configure an OSPF routing process. Specify the process ID to configure multiple instances of OSPF. A process ID is not needed of running a single OSPF instance.
Command Mode	Global Configuration
Syntax	router ospf router ospf <1-65535>
Parameters	**<1-65535>**: Process ID; unique for each routing process.
Example usage	`switch_a(config)# router ospf 100` `switch_a(config-router)#`

area authentication	
Purpose	Enable authentication for an OSPF area. Setting up a Type 1 authentication configures a 64-bit field for that particular network. All packets sent on this network must have this configured value in their OSPF header. This allows only routers that have the same passwords to join the routing domain. Use the **ip ospf authentication-key** command to specify a simple text password. Use the **ip ospf message-digest-key** to specify MD5 password.
Command Mode	Router Configuration
Syntax	[no] area (A.B.C.D | <0-4294967295>) authentication area (A.B.C.D | <0-4294967295>) authentication message-digest
Parameters	**A.B.C.D**: OSPF Area ID in IPv4 address format. **<0-4294967295>**: OSPF Area ID as 4-octet unsigned integer value. **message-digest**: Enable MD5 authentication in specified area ID.
Example usage	`switch_a(config-router)# area 1 authentication message-digest`

area default-cost	
Purpose	Specify a cost for the default summary route sent into a stub or NSSA area. This command provides the metric for the summary default route, generated by the area border router, into the NSSA or stub area. Use this option only on an area border router that is attached to the NSSA (Not-so-stubby area) or stub area.
Command Mode	Router Configuration
Syntax	area (A.B.C.D | <0-4294967295>) default-cost <0-16777215> no area (A.B.C.D | <0-4294967295>) default-cost
Parameters	**A.B.C.D**: OSPF Area ID in IPv4 address format. **<0-4294967295>**: OSPF Area ID as a decimal value. **default-cost**: Indicates the cost for the default summary route used for a stub or NSSA area. **<0-16777215>**: Stub's advertised default summary cost. Default is 1.
Example usage	`switch_a(config-router)# area 1 default-cost 10`

area filter-list	
Purpose	Configure a filter to advertise summary routes on an Area Border Router (ABR). This command suppresses incoming and outgoing summary routes between this area and other areas. You use this command in conjunction with the **prefix-list** and **access-list** commands.
Command Mode	Router Configuration
Syntax	area (A.B.C.D | <0-4294967295>) filter-list prefix WORD (in | out) area (A.B.C.D | <0-4294967295>) filter-list access WORD (in | out) no area (A.B.C.D | <0-4294967295>) filter-list prefix WORD (in | out) no area (A.B.C.D | <0-4294967295>) filter-list access WORD (in | out)
Parameters	**A.B.C.D**: OSPF area ID as an IPv4 address. **<0-4294967295>**: OSPF area ID as a decimal value. **prefix**: Use prefix list to filter summary. **WORD**: Name of the prefix or access list. **access**: Use access list to filter summary. **in**: Filter routes from other areas into this area. **out**: Filter routes from this area into other areas.

Example usage	`switch_a(config)#access-list 1 deny 172.22.0.0/8`
	`switch_a(config)#router ospf 100`
	`switch_a(config-router)#area 1 filter-list access 1 in`

area multi-area-adjacency	
Purpose	Enable multi-area adjacency on the specified interface. Multi-area adjacency establishes adjacency between the Area Border Routers (ABRs). The specified interface of the ABR is associated with multiple areas. Multiple OSPF interfaces must be created for multiple areas.
Command Mode	Router Configuration
Syntax	area (A.B.C.D | <0-4294967295>) multi-area-adjacency IFNAME neighbor A.B.C.D no area (A.B.C.D | <0-4294967295>) multi-area-adjacency IFNAME (neighbor A.B.C.D |)
Parameters	**IFNAME**: An alphanumeric string that is the interface name. neighbor: Set the neighbor. **A.B.C.D**: IP address of neighbor.

Example usage	`switch_a(config)#router ospf 1`
	`switch_a(config)#router-id 10.10.10.10`
	`switch_a(config-router)#area 1 multi-area-adjacency ge20 neighbor 20.20.20.10`

area range	
Purpose	Summarize OSPF routes at an area boundary. A single summary route is then advertised to other areas by the Area Border Routers (ABRs). Routing information is condensed at area boundaries and outside the area. If the network numbers in an area are assigned in a way such that they are contiguous, the ABRs can be configured to advertise a summary route that covers all the individual networks within the area that fall into the specified range.
Command Mode	Router Configuration
Syntax	area (A.B.C.D | <0-4294967295>) range A.B.C.D/M area (A.B.C.D | <0-4294967295>) range A.B.C.D/M advertise area (A.B.C.D | <0-4294967295>) range A.B.C.D/M not-advertise no area (A.B.C.D | <0-4294967295>) range A.B.C.D/M no area (A.B.C.D | <0-4294967295>) range A.B.C.D/M (advertise | not-advertise)
Parameters	**A.B.C.D**: OSPF Area ID in IPv4 address format. **<0-4294967295>**: OSPF Area ID as a decimal value. **A.B.C.D/M**: Area range prefix and length. **advertise**: Advertise this range. **not-advertise**: Do not advertise this range.
Example usage	`switch_a(config-router)# area 1 range 192.16.0.0/24`

area nssa	
Purpose	Set an area as a Not-So-Stubby-Area (NSSA). There are no external routes in an OSPF stub area, so you cannot redistribute from another protocol into a stub area. An NSSA allows external routes to be flooded within the area. These routes are then leaked into other areas. However, the external routes from other areas still do not enter the NSSA. You can configure an area to be a stub area or an NSSA, but not both. This command simplifies administration when connecting a central site using OSPF to a remote site that is using a different routing protocol. You can extend OSPF to cover the remote connection by defining the area between the central router and the remote router as a NSSA.
Command Mode	Router Configuration
Syntax	[no] area (A.B.C.D | <0-4294967295>) nssa area (A.B.C.D | <0-4294967295>) nssa (translate-candidate | translate-always) area (A.B.C.D | <0-4294967295>) nssa {translator-role (candidate | always) | stabilityinterval <0-2147483647> | no-redistribution | default-information-originate (metric <0-16777214> | metric-type <1-2> | metric <0-16777214> metric-type <1-2> | metric-type <1-2> metric <0-16777214> |) | no-summary} no area (A.B.C.D | <0-4294967295>) nssa {translator-role | no-redistribution | defaultinformation-originate | no-summary}
Parameters	**A.B.C.D**: OSPF Area ID in IPv4 address format. **<0-4294967295>**: OSPF Area ID as a decimal value. **translator-role**: NSSA-ABR translator role **candidate**: Translate NSSA-LSA to Type-5 LSA if router is elected. **always**: Always translate NSSA-LSA to Type-5 LSA. **stability-interval**: Stability timer for a NSSA area. If an elected translator determines its services are no longer required, it continues to perform its duties for this time interval. This minimizes excess flushing of translated Type-7 LSAs and provides a more stable translator transition. **<0-2147483647>:** Stability interval in seconds. **no-redistribution**: Do not redistribute into the NSSA. **default-information-originate**: Originate Type-7 default LSA into the NSSA. **metric**: Set metric for default routes. **<0-16777214>:** Metric value.
Example usage	`switch_a(config-router)# area 3 nssa translator-role candidate noredistribution default-information-originate metric 34 metric-type 2`

area shortcut	
Purpose	Configure the short-cutting mode of an area. An area shortcut enables traffic to go through the non-backbone area with a lower metric whether or not an ABR router is attached to the backbone area.
Command Mode	Router Configuration
Syntax	area (A.B.C.D | <0-4294967295>) shortcut (default | enable | disable) no area (A.B.C.D | <0-4294967295>) shortcut no area (A.B.C.D | <0-4294967295>) shortcut (enable | disable)
Parameters	**A.B.C.D**: OSPF Area ID in IPv4 address format. **<0-4294967295>**: OSPF Area ID as a decimal value. **default**: Sets default short-cutting behavior. **enable**: Forces short-cutting through the area. **disable**: Disables short-cutting through the area.
Example usage	`switch_a(config-router)# area 1 shortcut default`

area stub	
Purpose	Define an area as a stub area. There are two stub area router configuration commands: the **stub** and **default-cost** commands. In all routers attached to the stub area, configure the area by using the stub option of the area command. For an area border router (ABR) attached to the stub area, use the **area default-cost** command. Use the **no-summary** parameter with this command to define a totally stubby area. Define an area as a totally stubby area when routers in the area do not need to learn about summary LSAs from other areas.
Command Mode	Router Configuration
Syntax	area (A.B.C.D | <0-4294967295>) stub area (A.B.C.D | <0-4294967295>) stub no-summary no area (A.B.C.D | <0-4294967295>) stub no area (A.B.C.D | <0-4294967295>) stub no-summary
Parameters	**A.B.C.D**: OSPF Area ID in IPv4 address format. **<0-4294967295>**: OSPF Area ID as a decimal value. **no-summary**: Stops an ABR from sending summary link advertisements into the stub area.
Example usage	`switch_a(config-router)# area 1 stub no-summary`

area virtual-link	
Purpose	Configure a link between two backbone areas that are physically separated through other nonbackbone area. Configure the **hello-interval** to be the same for all routers attached to a common network. A short **hello-interval** results in the router detecting topological changes faster but also increases routing traffic. The **retransmit-interval** is the expected round-trip delay between any two routers in a network. Set the value to be greater than the expected round-trip delay to avoid needless retransmissions. The **transmit-delay** is the time taken to transmit a link state update packet on the interface.
Command Mode	Router Configuration
Syntax	[no] area (A.B.C.D | <0-4294967295>) virtual-link A.B.C.D area (A.B.C.D | <0-4294967295>) virtual-link A.B.C.D {authentication (messagedigest | null|) | authentication-key LINE | message-digest-key <1-255> md5 LINE | deadinterval <1-65535> | hello-interval <1-65535> | retransmit-interval <1-3600> | transmit-delay <1-3600>} [no] area (A.B.C.D | <0-4294967295>) virtual-link A.B.C.D {fall-over bfd} no area (A.B.C.D |<0-4294967295>) virtual-link A.B.C.D {dead-interval | hellointerval | retransmit-interval | transmit-delay | authentication | authenticationkey | message-digest-key <1-255>}
Parameters	**A.B.C.D**: OSPF Area ID in IPv4 address format. **<0-4294967295>**: OSPF Area ID as a decimal value. **A.B.C.D**: IP address of the virtual link neighbor. **message-digest**: Cryptographic authentication. **null**: Null authentication. **authentication-key:** Set authentication key. **LINE**: Authentication key ID of 8 characters. **<1-255>**: Message digest key. **md5**: Set the MD5 key - **LINE**: MD5 key. **dead-interval**: Interval during which no packets are received and after which the router acknowledges a neighboring router as off-line. **<1-65535>:** The interval in seconds. Default is 40. **hello-interval**: Interval the router waits before it sends hello packet. <1-65535> in seconds. Default is 10 seconds. **retransmit-interval**: Interval router waits before it retransmits a packet. <1-3600> in seconds. Default is 5 seconds. **transmit-delay**: Interval router waits before it transmits a packet. <1-3600> in seconds. Default is 1 second. **fall-over**: Specify fall-over detection. **bfd**: Bidirectional Forwarding Detection (BFD)
Example usage	`switch_a(config-router)# area 1 virtual-link 10.10.11.50 hello 5 dead 10`

auto-cost reference bandwidth	
Purpose	Control how OSPF calculates the default metric for the interface. By default, OSPF calculates the OSPF metric for an interface by dividing the reference bandwidth by the interface bandwidth. The default value for the reference bandwidth is 100Mbps. The auto-cost command is used to differentiate high bandwidth links. For multiple links with high bandwidth, specify a larger reference bandwidth value to differentiate cost on those links.
Command Mode	Router Configuration
Syntax	auto-cost reference-bandwidth <1-4294967> no auto-cost reference-bandwidth
Parameters	**<1-4294967>**: Reference bandwidth in Mbps. Default is 100 Mbps.
Example usage	`switch_a(config-router)# auto-cost reference-bandwidth 50`

bfd all-interfaces	
Purpose	Enable Bidirectional Forwarding Detection (BFD) on all interfaces.
Command Mode	Router Configuration
Syntax	[no] bfd all-interfaces
Parameters	None
Example usage	`switch_a(config-router)# bfd all-interfaces reference-bandwidth 50`

capability opaque	
Purpose	Enable opaque-LSAs which are Type 9, 10 and 11 LSAs that deliver information used by external applications.
Command Mode	Router Configuration
Syntax	[no] capability opaque
Parameters	None
Example usage	`switch_a(config-router)# compatibility opaque`

capability restart	
Purpose	Enable OSPF graceful restart or restart signaling. If a router is not restart-enabled, it cannot enter graceful restart mode and act as a helper.
Command Mode	Router Configuration
Syntax	capability restart (graceful | signaling) no capability restart
Parameters	None
Example usage	`switch_a(config-router)# capability restart graceful`

compatible rfc1583	
Purpose	Restore the method used to calculate summary route costs per RFC. RFC 1583 specifies a method for calculating the metrics for summary routes based on the minimum metric of the component paths available. RFC 2328 specifies a method for calculating metrics based on maximum cost. With this change, it is possible that all of the ABRs in an area might not be upgraded to the new code at the same time. This command addresses this issue and allows the selective disabling of RFC 2328 compatibility.
Command Mode	Router Configuration
Syntax	[no] compatible rfc1583
Parameters	None
Example usage	`switch_a(config-router)# compatible rfc1583`

default-information originate	
Purpose	Create a default external route into an OSPF routing domain. The system acts like an Autonomous System Boundary Router (ASBR) when you use the **default-information originate** command to redistribute routes into an OSPF routing domain. An ASBR does not by default generate a default route into the OSPF routing domain. When you give the **default-information originate** command, also specify a **route-map** to avoid a dependency on the default network in the routing table.
Command Mode	Router Configuration
Syntax	default-information originate default-information originate {metric <0-16777214> | metric-type (1 | 2) | {route-map WORD | always} no default-information originate no default-information originate {metric | metric-type | {route-map | always}
Parameters	**always**: Used to advertise the default route regardless of whether there is a default route. **metric**: Sets the OSPF metric used in creating the default route. **<0-16777214>**: Sets the OSPF metric used in creating the default route. Default metric value is 10. **metric-type**: The external link type associated with the default route advertised into the OSPF routing domain (see RFC 3101). **1**: Sets OSPF External Type 1 metric. **2**: Sets OSPF External Type 2 metric (default). **route-map**: Route map. **WORD**: Name of route map.
Example usage	`switch_a(config-router)# default-information originate always metric 23 metrictype 2 route-map myinfo`

default-metric	
Purpose	Set a default metric for OSPF. A default metric facilitates redistributing routes with incompatible metrics. If the metrics do not convert, the default metric provides an alternative. Use this command to use the same metric value for all redistributed routes. Use this command in conjunction with **redistribute**.
Command Mode	Router Configuration
Syntax	default-metric <1-16777214> no default-metric
Parameters	**<1-16777214>**: Metric value
Example usage	`switch_a(config-router)# default-metric 100`

distance	
Purpose	Set OSPF administrative distances. The administrative distance rates the trustworthiness of a routing information source. A higher distance value means a lower trust rating. Use the no form to restore the default value (110).
Command Mode	Router Configuration
Syntax	[no] distance <1-255> [no] distance <1-255> A.B.C.D/M (WORD |) distance ospf {intra-area <1-255> | inter-area <1-255> | external <1-255>} no distance ospf
Parameters	**<1-255>**: Default administrative distance to be used. **intra-area:** Routes within an area, **<1-255>**: Distance for all routes within an area **inter-area**: Routes from one area to another area. **<1-255>**: Distance for all routes from one area to another area. **external**: Routes from other domains learned by redistribution. **<1-255>**: Distance from other domains learned by redistribution. **A.B.C.D/M**: Distance for routes to prefixes whose nexthop matches this address. **WORD**: Name of access list to apply to route updates.
Example usage	`switch_a(config-router)# distance ospf inter-area 20 intra-area 10 external 40`

distribute-list	
Purpose	Filter networks in routing updates. This command redistributes other routing protocols into the OSPF routing table.
Command Mode	Router Configuration
Syntax	[no] distribute-list WORD out (kernel | connected | static | rip | bgp | isis | ospf (<1-65535> |)) [no] distribute-list WORD in
Parameters	**WORD**: Name of the access list. **in, out**: Filter incoming and outgoing routing updates. **kernel**: Specify kernel routes. **connected**: Specify connected routes. **<1-65535>**: OSPF process ID. If not specified, this command redistributes all running OSPF processes.

Example usage	`switch_a(config)#access-list list1 permit 172.10.0.0/16`
	`switch_a(config)#router ospf 100`
	`switch_a(config-router)#distribute-list list1 out bgp`

enable db-summary-opt	
Purpose	Enable the database summary list optimization for OSPFv2. When enabled, the database exchange process is optimized by removing the LSA from the database summary list for the neighbor, if the LSA instance in database summary list is the same as or less recent than the listed LSA in the database description packet received from the neighbor.
Command Mode	Router Configuration
Syntax	[no] enable db-summary-opt
Parameters	None
Example usage	`switch_a(config-router)# enable db-summary-opt`

host area	
Purpose	Configure a stub host entry belonging to a particular area. Using this command, you can advertise specific host routes in the router-LSA as stub link. Since stub host belongs to the specified router, specifying cost is not important.
Command Mode	Router Configuration
Syntax	host A.B.C.D area (A.B.C.D | <0-4294967295>) host A.B.C.D area (A.B.C.D | <0-4294967295>) cost <0-65535> no host A.B.C.D area (A.B.C.D | <0-4294967295>) no host A.B.C.D area (A.B.C.D | <0-4294967295>) cost (<0-65535> |)
Parameters	**A.B.C.D**: IP address of the host. **area**: Set the OSPF area ID **A.B.C.D**: OSPF Area ID in IPv4 address format. **<0-4294967295>**: OSPF Area ID as a decimal value. **cost**: Specify cost for stub host entry. **<0-65535>**: Cost for stub host entry.
Example usage	`switch_a(config-router)# host 172.16.10.101 area 2 cost 10`

max-concurrent-dd	
Purpose	Limit the number of Database Descriptors (DD) that can be processed concurrently. This command limits the maximum number of DD exchanges that can occur concurrently per OSPF instance, thus allowing for all of the adjacencies to come up.
Command Mode	Router Configuration
Syntax	max-concurrent-dd <1-65535> no max-concurrent-dd
Parameters	**<1-65535>**: Number of DD processes.
Example usage	`switch_a(config-router)# max-concurrent-dd 4`

maximum-area	
Purpose	Configure the maximum number of OSPF areas.
Command Mode	Router Configuration
Syntax	maximum-area <1-4294967294> no maximum-area
Parameters	**<1-4294967294>**: Maximum number of OSPF areas.
Example usage	`switch_a(config-router)# maximum-area 5000`

neighbor	
Purpose	Configure OSPF routers interconnecting to NBMA (Non-Broadcast Multi-Access) networks. Include one neighbor entry for each known non-broadcast network neighbor. Configure the neighbor address on the primary address of the interface.
Command Mode	Router Configuration
Syntax	[no] neighbor A.B.C.D [no] neighbor A.B.C.D (priority <0-255> | poll-interval <1-2147483647> | cost <1-65535>) [no] neighbor A.B.C.D (cost <1-65535>)
Parameters	**priority**: Router priority of the non-broadcast neighbor associated with the specified IP address - <0-255>, default is 0. **poll-interval**: Rate at which routers send hello packets when neighboring router inactive, <1-2147483647> in seconds. Set this value much larger than hello interval. Default is 120. **cost**: Link-state metric to this neighbor, <1-65535>.
Example usage	`switch_a(config-router)# neighbor 1.2.3.4 priority 1 poll-interval 90`

network	
Purpose	Enable OSPF routing with a specified area ID (and optionally an instance ID) on interfaces with IP addresses that match the specified network address. OSPF routing is enabled per IPv4 subnet basis. You define the network address using the prefix length or a subnet mask. Use the no parameter with this command to disable OSPF routing on the interfaces.
Command Mode	Router Configuration
Syntax	**Network address defined using the prefix length:** network A.B.C.D/M area (A.B.C.D | <0-4294967295>) (instance-id <0-255> |) no network A.B.C.D/M area (A.B.C.D | <0-4294967295>) (instance-id <0-255>|) **Network address defined using subnet mask:** network A.B.C.D A.B.C.D area (A.B.C.D | <0-4294967295>) (instance-id <0-255> |) no network A.B.C.D A.B.C.D area (A.B.C.D |<0-4294967295>) (instance-id <0-255> |)
Parameters	**A.B.C.D/M**: IPv4 network address with prefix length. **A.B.C.D**: IPv4 network address. **A.B.C.D**: Subnet mask where the bits on left side are set to 1 to represent the network part and the bits on the right side are set to 0 to represent the host part. **area**: Set the OSPF area ID **A.B.C.D**: OSPF area ID in IPv4 address format. **<0-4294967295>:** OSPF area ID as a decimal value. instance-id: Instance ID. **<0-255>:** Instance ID. The default is 0.
Example usage	`switch_a(config-router)# network 10.0.0.0/8 area 1.1.1.1`

ospf abr-type	
Purpose	Set an OSPF Area Border Router (ABR) type. Use the no parameter to revert the ABR type to the default setting (Cisco). Specifying the ABR type allows better functioning in a multi-vendor environment. The ABR types are: • Cisco (RFC 3509): A router is considered an ABR if it has more than one area actively attached and one of them is the backbone area. • IBM (RFC 3509): A router is considered an ABR if it has more than one area actively attached and the backbone area is configured. In this case the configured backbone need not be actively connected. • Standard (RFC 2328): A router is considered an ABR if it has more than one area actively attached to it. • Shortcut (draft-ietf-ospf-shortcut-abr-02): This improves the standard ABR by modifying the calculation of interarea routes which are installed in non-backbone areas if the non-backbone path is better, thus providing a “shortcut” through these areas. To prevent routing loops, the inter-area routes are re-advertised only if they are associated with the backbone area.
Command Mode	Router Configuration
Syntax	ospf abr-type (cisco | ibm | standard | shortcut) no ospf abr-type (cisco | ibm | standard | shortcut |)
Parameters	**cisco**: Alternative ABR using Cisco implementation. This is the default ABR type. **ibm**: SAlternative ABR using IBM implementation. **standard**: Standard ABR. **shortcut**: Shortcut ABR.
Example usage	`switch_a(config-router)# ospf abr-type ibm`

ospf flood-reduction	
Purpose	Enable/disable flood reduction, which reduces unnecessary refreshing and flooding of already known and unchanged information.
Command Mode	Router Configuration
Syntax	[no] ospf flood-reduction
Parameters	None
Example usage	`switch_a(config-router)# ospf flood-reduction`

ospf router-id	
Purpose	Specify a router ID for the OSPF process. Configure each router with a unique router ID. In an OSPF router process which has active neighbors, a new router ID is used at the next reload or when you start the OSPF manually. Use the no parameter to force OSPF to use the previous router ID.
Command Mode	Router Configuration
Syntax	[no] ospf router-id A.B.C.D
Parameters	**A.B.C.D**: The router ID in IPv4 address format.
Example usage	`switch_a(config-router)# ospf router-id 2.3.4.5`

overflow database	
Purpose	Limit the maximum number of LSAs that can be supported by the OSPF instance. Use no parameter for an unlimited number of LSAs.
Command Mode	Router Configuration
Syntax	overflow database <0-4294967294> (hard | soft |) no overflow database
Parameters	**<0-4294967294>**: Maximum number of LSAs **hard**: Shutdown occurs if the number of LSAs exceeds the specified value. **soft**: Warning message appears if the number of LSAs exceeds the specified value.
Example usage	`switch_a(config-router)# overflow database 100 hard`

overflow database external	
Purpose	Limit the number of AS-external-LSAs a router can receive once it is in the wait state.
Command Mode	Router Configuration
Syntax	overflow database external <0-2147483647> <0-65535> no overflow database external
Parameters	**<0-2147483647>**: Maximum number of LSAs. This value should be the same on all routers in the AS. **<0-65535>**: Number of seconds the router waits before trying to exit the database overflow state. If this parameter is 0, the router exits the overflow state only after an explicit administrator command.
Example usage	`switch_a(config-router)# overflow database external 5 30`

passive-interface	
Purpose	Suppress sending Hello packets on all interfaces or on a specified interface. This command configures OSPF on simplex Ethernet interfaces. Since a simplex interface represents only one network segment between two devices, configure the transmitting interface as a passive interface. This ensures that OSPF does not send hello packets for the transmitting interface. Both the devices can see each other via the hello packet generated for the receiving interface.
Command Mode	Router Configuration
Syntax	[no] passive-interface IFNAME [no] passive-interface (IFNAME | A.B.C.D)
Parameters	**IFNAME**: The name of the interface. **A.B.C.D**: IP address of the interface.
Example usage	`switch_a(config-router)# passive-interface ge10`

redistribute	
Purpose	Redistribute routes from a routing protocol, static route, and kernel route into an OSPF routing table.
Command Mode	Router Configuration
Syntax	redistribute (kernel | connected | static | rip | bgp | isis | ospf (<1-65535> |)) {metric <0-16777214> | metric-type (1 | 2) | route-map WORD | tag <0-4294967295>} no redistribute (kernel | connected | static | rip | bgp | isis | ospf (<1-65535> |)) {metric | metric-type | route-map | tag}
Parameters	**kernel**: Kernel routes. **connected**: Connected routes. **ospf**: OSPF instance to redistribute a particular OSPF instance into another OSPF instance. **<1-65535>**: OSPF process ID **metric**: Specify the external metric, <0-16777214> **metric-type**: External metric-type **1**: OSPF External Type 1 metrics. **2**: OSPF External Type 2 metrics. **route-map**: Specify a route map reference. **WORD**: Name of the route-map. **tag**: Tag value to use as a “match” value for controlling redistribution via route maps **<0-4294967295>**: Route tag.
Example usage	`switch_a(config-router)# redistribute bgp metric 12`

router-id	
Purpose	Specify a router ID for the OSPFv3 process. Configure each router with a unique router-id. In an OSPFv3 router process that has active neighbors, a new router-id is used at the next reload or when you start the OSPFv3 manually. Use the no form to force OSPFv3 to stop routing functionality
Command Mode	Router Configuration
Syntax	router-id A.B.C.D no router-id (A.B.C.D |)
Parameters	**A.B.C.D:** The router ID in IPv4 address format.
Example usage	`switch_a(config-router)# router-id 32.53.4.5`

summary-address	
Purpose	Summarize or suppress external routes with the specified address range. An address range is a pairing of a starting address and a mask that is almost the same as IP network number. Redistributing routes from other protocols into OSPF requires the router to advertise each route individually in an external LSA. Use this command to advertise one summary route for all redistributed routes covered by a specified network address and mask. This minimizes the size of the OSPF link state database.
Command Mode	Router Configuration
Syntax	summary-address X:X::X:X/M (not-advertise | tag <0-4294967295>|) summary-address A.B.C.D/M (not-advertise | tag <0-4294967295>|) no summary-address X:X::X:X/M no summary-address A.B.C.D/M no summary-address X:X::X:X/M (not-advertise | tag (<0-4294967295> |)) no summary-address A.B.C.D/M (not-advertise | tag (<0-4294967295> |))
Parameters	**X:X::X:X/M**: The range of addresses given as IPv6 starting address and a mask. **A.B.C.D/M**: The range of addresses given as IPv4 starting address and a mask. **not-advertise**: Suppress routes that match the range. **tag**: Tag value to use as a "match" value for controlling redistribution via route maps. **<0-4294967295>**: Tag value. The default is 0.
Example usage	`switch_a(config-router)# summary-address 10.10.10.0/24 not-advertise`

timers lsa arrival	
Purpose	Set the minimum interval to accept the same link-state advertisement (LSA) from OSPF neighbors. Use the no form to restore the default value (1000 milliseconds).
Command Mode	Router Configuration
Syntax	timers lsa arrival <0-600000> no timers lsa arrival
Parameters	**<0-600000>**: Minimum delay in milliseconds between accepting the same LSA from neighbors.
Example usage	`switch_a(config-router)# timers lsa arrival 10000`

timers throttle lsa	
Purpose	Sets the rate-limiting intervals for OSPF link-state advertisement (LSA) generation. Use the no form to restore default values.
Command Mode	Router Configuration
Syntax	timers throttle lsa all <0-600000> <1-600000> <1-600000> no timers throttle lsa all
Parameters	**<0-600000>: Start interval** - The minimum delay in milliseconds for the generation of LSAs. The first instance of LSA is always generated immediately upon a local OSPF topology change. The generation of the next LSA is not before the start interval. **<0-600000>: Hold interval** - The hold time in milliseconds. This value is used to calculate the subsequent rate limiting times for LSA generation. **<0-600000>: Maximum interval** - The maximum wait time in milliseconds between generation of the same LSA.
Example usage	`switch_a(config-router)# timers throttle lsa all 200 10000 45000`
Note	Default values: Start interval: 0 milliseconds Hold interval: 5000 milliseconds Maximum interval: 5000 milliseconds

timers spf	
Purpose	Adjust route-calculation timers. This command configures the delay time between the receipt of a topology change and the calculation of the Shortest Path First (SPF). This command also configures the hold time between two consecutive SPF calculations.
Command Mode	Router Configuration
Syntax	timers spf exp <0-2147483647> <0-2147483647> no timers spf exp
Parameters	**<0-2147483647>**: Minimum delay between receiving a change to SPF calculation (in milliseconds). **<0-2147483647>**: Maximum delay between receiving a change to SPF calculation (in milliseconds).
Example usage	`switch_a(config-router)# timers spf exp 10000 25000`

OSPF Interface Commands

ip ospf authentication	
Purpose	Send and receive OSPF packets with the specified authentication method on the current interface.
Command Mode	Interface Configuration
Syntax	ip ospf authentication (null | message-digest |) ip ospf A.B.C.D authentication (null | message-digest |) no ip ospf (A.B.C.D |) authentication
Parameters	**A.B.C.D**: The IP address of the interface. **null**: Use no authentication. **message-digest**: Message digest authentication.
Example usage	`switch_a(config-if)# ip ospf authentication null`

ip ospf authentication-key	
Purpose	Specify an OSPF authentication password for neighboring routers. This command creates a password (key) that is inserted into the OSPF header when the switch software originates packets. Assign a separate password to each network for different interfaces. All neighboring routers on the same network with the same password exchange OSPF routing data. The key can be used only when authentication is enabled for an area with the area authentication command. Simple password authentication allows a password to be configured for each area. Configure the routers in the same routing domain with the same password.
Command Mode	Interface Configuration
Syntax	ip ospf (A.B.C.D|) authentication-key LINE no ip ospf (A.B.C.D|) authentication-key
Parameters	**A.B.C.D**: The IP address of the interface. **LINE**: Authentication password.
Example usage	Create an authentication key **testkey** on interface ge24 in area 0.

	`switch_a#configure terminal`
	`switch_a(config)#router ospf 100`
	`switch_a(config-router)#network 10.10.10.0/24 area 0`
	`switch_a(config-router)#area 0 authentication`
	`switch_a(config-router)#exit`
	`switch_a(config)#interface ge24`
	`switch_a(config-if)#ip ospf 12.10.10.2 authentication-key testkey`

ip ospf cost	
Purpose	Specify the cost of the link-state metric in a router-LSA. The interface cost indicates the overhead required to send packets across an interface. This cost is stated in the Router-LSA's link. The cost is inversely proportional to the bandwidth of an interface. By default, the cost of an interface is calculated based on the bandwidth (108/ bandwidth). Use this command to set the cost manually. Use the no parameter to reset the cost to its default value.
Command Mode	Interface Configuration
Syntax	ip ospf (A.B.C.D |) cost <1-65535> no ip ospf (A.B.C.D |) cost
Parameters	<1-65535>: The link-state metric. Default value is 10.
Example usage	`switch_a(config-if)# ip ospf 10.10.12.12 cost 200`

ip ospf database-filter	
Purpose	Turn on the LSA database-filter for a particular interface. By default, OSPF floods new LSAs over all interfaces in an area, except the interface on which the LSA was received. Too much flooding wastes bandwidth and can lead to excessive link and CPU usage in certain topologies. To avoid this, you can block flooding of LSAs over specified interfaces.
Command Mode	Interface Configuration
Syntax	ip ospf (A.B.C.D|) database-filter all out no ip ospf (A.B.C.D|) database-filter
Parameters	**A.B.C.D**: IP address of the interface.
Example usage	`switch_a(config-if)# ip ospf database-filter all out`

ip ospf dead-interval	
Purpose	Set the interval during which the router waits to receive an OSPF hello packet from the neighbor before declaring the neighbor down. This value is advertised in the router's hello packets. This value must be a multiple of **hello-interval** and be the same for all routers on the network.
Command Mode	Interface Configuration
Syntax	ip ospf (A.B.C.D |) dead-interval <1-65535> no ip ospf (A.B.C.D |) dead-interval
Parameters	**<1-65535>**: Interval in seconds. Default is 40 seconds.
Example usage	`switch_a(config-if)# ip ospf dead-interval 100`

ip ospf disable	
Purpose	Disable OSPF packet processing on an interface. This command overrides the **network area** command.
Command Mode	Interface Configuration
Syntax	ip ospf disable all no ip ospf disable all
Parameters	None
Example usage	`switch_a(config-if)# ip ospf disable all`

ip ospf flood-reduction	
Purpose	Enable/disable flood reduction on an interface. This reduces unnecessary refreshing and flooding of known and unchanged information.
Command Mode	Interface Configuration
Syntax	[no] ip ospf flood-reduction
Parameters	None
Example usage	`switch_a(config-if)# ip ospf flood-reduction`

ip ospf hello-interval	
Purpose	Set the interval between hello packets. Configure the same hello-interval for all routers on a network. A shorter hello interval ensures faster detection of topological changes but results in more routing traffic.
Command Mode	Interface Configuration
Syntax	ip ospf (A.B.C.D |) hello-interval <1-65535> no ip ospf (A.B.C.D |) hello-interval
Parameters	**<1-65535>**: Interval in seconds. Default is 10 seconds.
Example usage	`switch_a(config-if)# ip ospf hello-interval 10`

ip ospf message-digest-key	
Purpose	Register an MD5 key for OSPF authentication. Use the no parameter to remove an MD5 key. Message Digest Authentication is cryptographic authentication. A key (password) and key-id are configured on each router. The router uses an algorithm based on the OSPF packet, the key, and the key-id to generate a message digest that is appended to the packet. Use this command for uninterrupted transitions between passwords. This is helpful for administrators who want to change the OSPF password without disrupting communication. The system begins a rollover process until all the neighbors have adopted the new password. This allows neighboring routers to continue communication while they are being updated with the new password. The router will stop sending duplicate packets once it detects that all neighbors have adopted the new password. Maintain only one password per interface, removing the old password when you add a new one. This prevents the local system from continuing to communicate with the system that is using the old password. Removing the old password also reduces overhead during rollover. All neighboring routers on the same network must have the same password value to enable exchange of OSPF routing data.
Command Mode	Interface Configuration
Syntax	ip ospf (A.B.C.D |) message-digest-key <1-255> md5 LINE no ip ospf (A.B.C.D |) message-digest-key <1-255>
Parameters	**A.B.C.D**: IPv4 address of the interface. **message-digest-key**: Specify a key ID. **<1-255>**: Key ID. **md5**: Specify a key (password). **LINE**: The OSPF password (1-16 characters).
Example usage	`switch_a(config-if)#ip ospf authentication message-digest` `switch_a(config-if)#ip ospf message-digest-key 1 md5 passwordsample`

ip ospf mtu	
Purpose	Set MTU size for OSPF to construct packets based on this value. Whenever OSPF constructs packets, it uses interface MTU size as Maximum IP packet size. This command forces OSPF to use the specified value overriding the actual interface MTU size. This command does not configure the MTU settings in the kernel. OSPF does not recognize MTU size changes made in the kernel until the MTU size is updated through this command.
Command Mode	Interface Configuration
Syntax	ip ospf mtu <576-65535> no ip ospf mtu
Parameters	**<576-65535>**: MTU size.
Example usage	`switch_a(config-if)# ip ospf mtu 10000`

ip ospf mtu-ignore	
Purpose	Configure OSPF so that it does not check the MTU size during DD (Database Description) exchange.
Command Mode	Interface Configuration
Syntax	ip ospf (A.B.C.D|) mtu-ignore no ip ospf (A.B.C.D|) mtu-ignore
Parameters	**A.B.C.D**: IP address of the interface.
Example usage	`switch_a(config-if)# ip ospf mtu-ignore`

ip ospf network	
Purpose	Set the OSPF network type. Use the no parameter to return to the default value (Broadcast).
Command Mode	Interface Configuration
Syntax	ip ospf network (broadcast | non-broadcast | point-to-multipoint | point-to-point) ip ospf network point-to-multipoint non-broadcast no ip ospf network
Parameters	**broadcast**: Set the network type to broadcast. **non-broadcast**: Set the network type to NBMA. **point-to-multipoint**: Set the network type to point-to-multipoint. **point-to-point**: Set the network type to point-to-point.
Example usage	`switch_a(config-if)# ip ospf network point-to-point`

ip ospf priority	
Purpose	Set the router priority to determine the designated router (DR) for the network. A router with the higher router priority becomes the DR. If the priority is the same for two routers, the router with the higher router ID takes precedence. The default priority is 1.
Command Mode	Interface Configuration
Syntax	ip ospf (A.B.C.D|) priority <0-255> no ip ospf (A.B.C.D|) priority
Parameters	**A.B.C.D**: IP address of the interface. **priority**: Specify the router priority of the interface. **<0-255>**: Router priority of the interface.
Example usage	`switch_a(config-if)# ip ospf priority 20`

ip ospf retransmit-interval	
Purpose	Specify the time between link-state advertisement (LSA) retransmissions for adjacencies belonging to the interface.
Command Mode	Interface Configuration
Syntax	ip ospf (A.B.C.D |) retransmit-interval <5-65535> no ip ospf (A.B.C.D |) retransmit-interval
Parameters	**A.B.C.D**: IP address of the interface. **<5-65535>**: Interval in seconds. Default is 5 seconds.
Example usage	`switch_a(config-if)# ip ospf retransmit-interval 20`

ip ospf transmit-delay	
Purpose	Set the estimated time it takes to transmit a link-state-update packet on the interface. The transmit delay value adds a specified time to the age field of an update. If the delay is not added, the time in which the LSA transmits over the link is not considered. This command is especially useful for low speed links. Add transmission and propagation delays when setting the transmit delay value. Use the no parameter to return to the default value 1 second).
Command Mode	Interface Configuration
Syntax	ip ospf (A.B.C.D |) transmit-delay <1-65535> no ip ospf (A.B.C.D |) transmit-delay
Parameters	**A.B.C.D**: IP address of the interface. **<1-65535>**: Time in seconds to transmit a link-state update
Example usage	`switch_a(config-if)# ip ospf transmit-delay 5`

20 VRRP (Virtual Router Redundancy Protocol)

VRRP Information

show vrrp	
Purpose	Show VRRP configuration data and statistics. VRRP provides for automatic assignment of available IP routers to participating hosts. This increases the availability and reliability of routing paths via automatic default gateway selections on an IP subnetwork.
Command Mode	Privileged exec
Syntax	show vrrp <1-255> <IFNAME> show vrrp statistics <1-255> <IFNAME>
Parameters	**<1-255>**: VRRP IPv4 router identifier **<IFNAME>**: Interface name
Example usage	`switch_a# show vrrp 10 ge1`

VRRP Configuration

router vrrp	
Purpose	Enter router mode to configure an OSPF routing process. VRRP sessions cannot be enabled on L2 interfaces.
Command Mode	Global Configuration
Syntax	router vrrp <1-255> <IFNAME>
Parameters	**<1-255>**: Virtual router identifier **IFNAME**: Interface name
Example usage	`switch_a(config)# router vrrp 1 ge23` `switch_a(config-router)#`

vrrp vmac	
Purpose	Enable or disable Virtual MAC (VMAC). This command affects all VRRP groups in a router. On a single network segment, multiple VRRP groups can be configured, each using a different VMAC. The use of VMAC addressing allows for faster switchover when a backup router assumes the master role.
Command Mode	Global Configuration
Syntax	vrrp vmac (enable | disable)
Parameters	None
Example usage	`switch_a(config)# vrrp vmac enable`

accept-mode	
Purpose	Set accept mode for the session. Default is no accept-mode true.
Command Mode	Router Configuration
Syntax	accept-mode [true | false]
Parameters	None
Example usage	`switch_a(config-router)# accept-mode true`

advertisement-interval	
Purpose	Configure the advertisement interval of a virtual router. This is the length of time in seconds between each advertisement sent from the master to its backup(s). The master virtual router sends VRRP advertisements to other VRRP routers in the same group. The advertisements communicate the priority and state of the master virtual router.
Command Mode	Router Configuration
Syntax	advertisement-interval <1-255> no advertisement-interval
Parameters	**<1-255>**: Interval in seconds. Default is 1 second.
Example usage	`switch_a(config-router)# advertisement-interval 50`
Note	VRRP Master router and backup routers should be configured with the same advertisement interval. If there is a mismatch in the configuration, VRRP goes to the INIT state.

circuit-failover	
Purpose	Enable the VRRP circuit failover feature. If an electrical connection fails, the failover event will cause a reduction in VRRP priority by the configured delta in this command
Command Mode	Router Configuration
Syntax	[no] circuit-failover IFNAME <1-253> no circuit-failover (IFNAME |)
Parameters	**IFNAME**: Interface of the router that is monitored by the virtual router, usually an upstream interface. If the interface goes down, a router configured as backup may take over as a master. **<1-253>**: Delta value. The value by which the virtual router decrements its priority value during a circuit failover event. Configure this value to be greater than the difference of priorities between the master and backup routers.
Example usage	`switch_a(config-router)# circuit-failover ge10 200`

disable	
Purpose	Disable a VRRP session on the router. This command will cause a backup router to assume the role of master.
Command Mode	Router Configuration
Syntax	disable
Parameters	None
Example usage	`switch_a(config-router)# disable`

enable	
Purpose	Enable a VRRP session on the router. To make changes to the VRRP configuration, first disable the Router from participating in Virtual Routing using the **disable** command.
Command Mode	Router Configuration
Syntax	enable
Parameters	None
Example usage	`switch_a(config-router)# disable`
Note	Configure the virtual IP address and define an interface for the VRRP session (using the virtual-ip and interface commands) before enabling VRRP on a router.

preempt-mode	
Purpose	Configure preempt mode. When enabled (True), the highest priority backup is always the master when the default master is unavailable. If disabled (False), a higher priority backup will not preempt a lower priority backup that is acting as master. If the master router fails, the backup routers come online in priority order — highest to lowest. Preempt mode will cause a higher priority backup router to relieve a lower priority backup. By default, a preemptive scheme is enabled whereby a higher priority backup virtual router that becomes available takes over for the backup virtual router that was elected to become master virtual router. This preemptive scheme can be disabled using the preempt-mode false command. If preemption is disabled, the backup virtual router that is currently elected as Master does not transition to backup again when the alternate backup router with higher priority becomes available.
Command Mode	Router Configuration
Syntax	preempt-mode true preempt-mode false
Parameters	None.
Example usage	`switch_a(config-router)# preempt-mode false`

priority	
Purpose	Configure the VRRP router priority within the virtual router. Priority determines the role that each VRRP router plays and what happens if the master virtual router fails. If a VRRP router owns the IP address of the virtual router and the IP address of the physical interface, this router functions as the master virtual router. Priority also determines whether a VRRP router functions as a backup virtual router and the order of ascendancy to becoming a master virtual router if the master virtual router fails. Configure the priority of each backup virtual router with a value of 1 through 254 using the priority command.
Command Mode	Router Configuration
Syntax	priority <1-255> no priority
Parameters	**<1-255>:** Priority value. Set this to 255 for the master router.
Example usage	`switch_a(config-router)# priority 255`

switch-back delay	
Purpose	Set a switch-back delay timer for the master VRRP router. This feature prevents the original master VRRP router from transitioning back to the master state after coming back online until the configured delay timer has expired.
Command Mode	Router Configuration
Syntax	switch-back-delay <1-500000> no switch-back-delay
Parameters	**<1-500000>:** Delay in milliseconds. Default is 0.
Example usage	`switch_a(config-router)# switch-back-delay 7000`

virtual ip	
Purpose	Set the virtual IP address for the VRRP virtual router either as VRRP Master or Backup. This is the IP address used by end hosts to address their default gateway. The VRRP Master (and Owner) of the Virtual IP address only responds to packets destined to the Virtual IP address (for example, ICMP packets destined to the Virtual IP address). VRRP Master (and Not Owner) of the Virtual IP address does not respond to packets destined to the Virtual IP address, but forwards packets with a VMAC as the destination address.
Command Mode	Router Configuration
Syntax	virtual-ip [A.B.C.D] (master | backup | owner) no virtual-ip
Parameters	**A.B.C.D**: Specify the virtual IP address of the interface that participates in virtual routing. **master**: Specify the default state of the VRRP router within the Virtual Router as master. For master, the router must own the Virtual IP address. The owner is the router that has the virtual router address as its physical interface address. **backup**: Specify the default state of the VRRP router within the Virtual Router as backup. **owner**: Specify the IP address as the owner.
Example usage	`switch_a(config-router)# virtual-ip 10.10.20.30 master`

21 GVRP (Generic VLAN Registration Protocol)

GVRP Information

show gvrp	
Purpose	Show GVRP configuration, statistics, timer, and finite state machine.
Command Mode	Privileged exec
Syntax	show gvrp configuration show gvrp machine show gvrp statistics show gvrp timer
Parameters	None
Example usage	`switch_a# show gvrp statistics`

clear gvrp	
Purpose	Clear GVRP statistics
Command Mode	Privileged exec
Syntax	clear gvrp statistics clear gvrp statistics all clear gvrp statistics bridge BRIDGE_NAME clear gvrp statistics IFNAME
Parameters	**BRIDGE_NAME**: Bridge name **IFNAME**: Interface name
Example usage	`switch_a# clear gvrp statistics`

GVRP Configuration

set gvrp	
Purpose	Enable (set) and disable (reset) GVRP globally for the default bridge instance. This command does not enable or disable GVRP in all ports of the bridge. After enabling GVRP globally, use the **set port gvrp** command to enable GVRP on individual ports of the bridge..
Command Mode	Global Configuration
Syntax	set gvrp <enable | disable> bridge BRIDGE_NAME
Parameters	**BRIDGE_NAME**: Name of bridge.
Example usage	`switch_a(config)# set gvrp enable bridge 10`

set gvrp dynamic-vlan-creation	
Purpose	Enable and disable dynamic VLAN creation for a specific bridge instance.
Command Mode	Global Configuration
Syntax	set gvrp dynamic-vlan-creation <enable | disable> bridge BRIDGE_NAME
Parameters	**BRIDGE_NAME**: Name of bridge.
Example usage	`switch_a(config)# set dynamic-vlan-creation gvrp enable bridge 10`

set gvrp applicant state	
Purpose	Set the GVRP applicant state to normal or active..
Command Mode	Global Configuration
Syntax	set gvrp applicant state [active | normal] IFNAME
Parameters	**active**: Active state **normal**: Normal state **IFNAME**: Name of the interface.
Example usage	`switch_a(config)# set gvrp applicant state active ge1`

set gvrp timer

Purpose	Set GVMRP timers for a specific interface.
Command Mode	Global Configuration
Syntax	set gvrp timer [join | leave | leaveall] TIMER_VALUE IF_NAME
Parameters	**join**: Timer for joining the group. **leave**: Timer for leaving a group. **leaveall**: Timer for leaving all groups. **TIMER_VALUE** : Timer value in hundredths of a second <1-65535> **IF_NAME**: Name of the interface
Example usage	`switch_a(config)# set gvrp timer leave 245 ge1`

set port gvrp	
Purpose	Enable and disable GVRP on a port or all ports in a bridge.
Command Mode	Global Configuration
Syntax	set port gvrp <enable | disable> <IF_NAME | all>
Parameters	**all**: All ports added to recently configured bridge. **IFNAME**: Interface name
Example usage	`switch_a(config)# set port gvrp enable all`

22 GMRP (Generic Multiple Registration Protocol)

GMRP Information

show gmrp	
Purpose	Show GMRP configuration, statistics, timer, and finite state machine.
Command Mode	Privileged exec
Syntax	show gmrp configuration show gmrp machine show gmrp statistics show gmrp timer
Parameters	None
Example usage	`switch_a# show gmrp statistics`

clear gmrp	
Purpose	Clear GMRP statistics
Command Mode	Privileged exec
Syntax	clear gmrp statistics all clear gmrp statistics all bridge BRIDGE_NAME clear gmrp statistics vlanid <1-4094> clear gmrp statistics vlanid <1-4094> bridge <1-32> clear gmrp dynamic-entry bridge BRIDGE_NAME
Parameters	**BRIDGE_NAME**: Bridge name **<1-4094>**: VLAN ID **<1-32>**: Bridge ID
Example usage	`switch_a# clear gmrp statistics all`

GMRP Configuration

set gmrp	
Purpose	Enable (set) and disable (reset) GMRP globally for the default bridge instance. This command does not enable or disable GMRP in all ports of the bridge. After enabling GMRP globally, use **the set port gmrp** command to enable GMRP on individual ports of the bridge.
Command Mode	Global Configuration
Syntax	set gmrp <enable | disable> bridge <1-32>
Parameters	**<1-32>**: Bridge group ID
Example usage	`switch_a(config)# set gvrp enable bridge 10`

set gmrp extended-filtering bridge	
Purpose	Enable or disable extended filtering on a bridge as per Table 8-7 of IEEE802.1Q-2003.
Command Mode	Global Configuration
Syntax	set gmrp extended-filtering <enable | disable> bridge BRIDGE NAME
Parameters	**BRIDGE_NAME**: Bridge name.
Example usage	`switch_a(config)# set gmrp extended-filtering enable bridge 1`

set gmrp fwdall	
Purpose	Set the GMRP forward all option for an interface. If this command is not used, the default setting is GMRP disabled
Command Mode	Global Configuration
Syntax	set gmrp fwdall <enable | disable> IF_NAME
Parameters	**IF_NAME**: Interface name
Example usage	`switch_a(config)# set gmrp fwdall enable ge7`

set gmrp registration	
Purpose	Set GMRP registration type for all ports for a given bridge.
Command Mode	Global Configuration
Syntax	set gmrp registration < normal | fixed | forbidden | restricted> IFNAME
Parameters	**normal**: Dynamic GMRP multicast registration and deregistration on the port. **fixed**: Multicast groups currently registered on the switch are applied to the port, but any subsequent registrations or deregistrations do not affect the port. Any registered multicast groups on the port are not deregistered based on the GARP timers. **forbidden**: All GMRP multicasts are deregistered, and prevent any further GMRP multicast registration on the port. **restricted**: Restricted registration
Example usage	`switch_a(config)# set gmrp registration normal ge1`

set gmrp timer	
Purpose	Set the values for the GMRP Join, Leave, and Leaveall timers for a specified bridge.
Command Mode	Global Configuration
Syntax	set gvrp timer [join | leave | leaveall] TIMER_VALUE IF_NAME
Parameters	**join**: Timer for joining the group. **leave**: Timer for leaving a group. **leaveall**: Timer for leaving all groups. **TIMER_VALUE**: Timer value in hundredths of a second <1-65535> **IF_NAME**: Name of the interface
Example usage	`switch_a(config)# set gmrp timer leave 245 ge1`

set port gmrp	
Purpose	Enable/disable GMRP on a particular port in all VLANs or all ports in a bridge. GMRP on a port cannot be enabled for all VLANs if GMRP has already been configured for a particular VLAN for the port.
Command Mode	Global Configuration
Syntax	set port gmrp <enable | disable> <IF_NAME | all> vlan VLANID
Parameters	all: All ports added to recently configured bridge. IFNAME: Interface name
Example usage	`switch_a(config)# set port gmrp enable all`

23 PIM (Protocol Independent Multicast)

PIM Information

show ip pim	
Purpose	Display PIM configuration and settings data.
Command Mode	Privileged Exec
Syntax	show ip pim bsr-router show ip pim interface show ip pim local-members show ip pim mroute show ip pim neighbor show ip pim nexthop show ip pim rp show ip pim rp-hash show ip pim vrf
Parameters	None

PIM Configuration

ip pim accept-register	
Purpose	Configure the ability to filter out multicast sources specified by the given access-list at the rendezvous point (RP), so that the RP will accept/refuse to perform the Register mechanism for the packets sent by the specified sources. By default, the RP accepts Register packets from all multicast sources.
Command Mode	Global Configuration
Syntax	ip pim accept-register list (<100-199> | <2000-2699> | WORD) ip pim (vrf NAME|) accept-register list (<100-199> | <2000-2699> | WORD) no ip pim accept-register no ip pim (vrf NAME |) accept-register
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **<100-199>**: IP extended access-list value **<2000-2699>**: IP extended access-list value in the expanded range **WORD**: Name of a standard access list
Example usage	`switch_a(config)# ip pim accept-register list 121`

ip pim anycast-rp	
Purpose	Configure the Anycast RP in the RP set.
Command Mode	Global Configuration
Syntax	ip pim anycast-rp A.B.C.D A.B.C.D ip pim (vrf NAME |) anycast-rp A.B.C.D A.B.C.D no ip pim anycast-rp A.B.C.D no ip pim anycast-rp A.B.C.D A.B.C.D no ip pim (vrf NAME |) anycast-rp A.B.C.D no ip pim (vrf NAME |) anycast-rp A.B.C.D A.B.C.D
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **A.B.C.D**: Unicast IP address of the Anycast RP set. An Anycast RP set is a collection of RPs in the same domain. **A.B.C.D**: Destination IP address where Register messages
Example usage	`switch_a(config)# ip pim anycast-rp 1.1.1.1 10.10.10.10`

ip pim bsr-candidate	
Purpose	Give the router the candidate BSR status using the specified IP address of the interface.
Command Mode	Global Configuration
Syntax	ip pim bsr-candidate IFNAME ip pim bsr-candidate IFNAME <0-32> ip pim bsr-candidate IFNAME <0-32> <0-255> ip pim (vrf NAME |) bsr-candidate IFNAME ip pim (vrf NAME |) bsr-candidate IFNAME <0-32> ip pim (vrf NAME |) bsr-candidate IFNAME <0-32> <0-255> no ip pim bsr-candidate (IFNAME |) no ip pim (vrf NAME |) bsr-candidate (IFNAME |)
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **IFNAME** Specify the name of the interface **<0-32>**: Hash mask length for RP selection **<0-255>**: Priority for a BSR candidate
Example usage	`switch_a(config)# ip pim bsr-candidate ge24 20 30`

ip pim cisco-register-checksum	
Purpose	Configure the option to calculate the register checksum over the whole packet. Use for inter-operation with older Cisco IOS versions.
Command Mode	Global Configuration
Syntax	ip pim cisco-register-checksum ip pim cisco-register-checksum group-list (<1-99> | <1300-1999> | WORD) ip pim (vrf NAME |) cisco-register-checksum ip pim (vrf NAME |) cisco-register-checksum group-list (<1-99> | <1300-1999> | WORD) no ip pim cisco-register-checksum no ip pim cisco-register-checksum group-list (<1-99> | <1300-1999> | WORD) no ip pim (vrf NAME |) cisco-register-checksum no ip pim (vrf NAME |) cisco-register-checksum group-list (<1-99> | <1300-1999> | WORD)
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **group-list**: Use this parameter to configure the option to calculate the register checksum over the whole packet on multicast groups specified by the access-list. **<1-99>**: IP standard access-list. **<1300-1999>**: IP access-list (expanded range). **WORD**: IP named standard access list.
Example usage	`switch_a(config)# ip pim cisco-register-checksum group-list 34`

ip pim crp-cisco-prefix	
Purpose	Use this command to interoperate with Cisco devices that conform to an earlierdraft standard. Some Cisco devices might not accept candidate RPs with a groupprefix number of zero. Note that the latest BSR specification prohibits sending RPadvertisements with prefix 0. RP advertisements for the default IPv4 multicastgroup range 224/4 are sent with a prefix of 1.
Command Mode	Global Configuration
Syntax	[no] ip pim crp-cisco-prefix
Parameters	None
Example usage	`switch_a(config)# ip pim crp-cisco-prefix`

ip pim ignore-rp-set-priority	
Purpose	This command is used to inter-operate with older Cisco IOS versions. It allows the RP-SET priority value to be ignored, and only the hashing mechanism for RP selection used.
Command Mode	Global Configuration
Syntax	ip pim ignore-rp-set-priority ip pim (vrf NAME |) ignore-rp-set-priority no ip pim ignore-rp-set-priority no ip pim (vrf NAME |) ignore-rp-set-priority
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance
Example usage	`switch_a(config)# ip pim ignore-rp-set-priority`

ip pim jp-timer	
Purpose	Set a PIM join/prune timer.
Command Mode	Global Configuration
Syntax	[no] ip pim jp-timer <1-65535> [no] ip pim (vrf NAME|) jp-timer <1-65535> no ip pim jp-timer
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **<1-65535>**: Value of the Join/Prune timer, in seconds
Example usage	`switch_a(config)# ip pim jp-timer 234`

ip pim register-rate-limit	
Purpose	Configure the rate of Register packets sent by this designated router (DR), in number of packets per second. The configured rate is per (S,G) state, and is not a system-wide rate.
Command Mode	Global Configuration
Syntax	ip pim register-rate-limit <1-65535> ip pim (vrf NAME|) register-rate-limit <1-65535> no ip pim register-rate-limit no ip pim (vrf NAME|) register-rate-limit
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **<1-65535>**: Number of packets sent per second
Example usage	`switch_a(config)# ip pim register-rate-limit 20000`

ip pim register-rp-reachability	
Purpose	Enable the RP reachability check for PIM Registers at the designated router.
Command Mode	Global Configuration
Syntax	ip pim register-rp-reachability ip pim (vrf NAME |) register-rp-reachability no ip pim register-rp-reachability no ip pim (vrf NAME |) register-rp-reachability
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance
Example usage	`switch_a(config)# ip pim register-rp-reachability`

ip pim register-source	
Purpose	Configure the source address of Register packets sent by this DR, overriding the default source address, which is the address of the RPF interface toward the source host. Use the no option to remove the source address of register packets sent by this DR, and reset it to use the default source address, that is, the address of the RPF interface toward the source host. The configured address must be a reachable address so the RP can send corresponding Register-Stop messages in response. This address is usually the loopback interface address, but can also be other physical addresses. The address must be advertised by unicast routing protocols on the DR.
Command Mode	Global Configuration
Syntax	ip pim register-source A.B.C.D ip pim register-source IFNAME ip pim (vrf NAME|) register-source A.B.C.D ip pim (vrf NAME|) register-source IFNAME no ip pim register-source no ip pim (vrf NAME|) register-source
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **A.B.C.D**: The IP address to use as the source of the register packets **IFNAME**: The name of the interface to use as the source of the register packets
Example usage	`switch_a(config)# ip pim register-source 3.3.3.2`
Note	The interface configured does not require PIM to be enabled.

ip pim register-suppression	
Purpose	Configure the register-suppression time, in seconds. Configuring this value modifies register-suppression time at the designated router; configuring this value at the rendezvous point modifies the RPkeepalive-period value if the ip pim rp-register-kat command is not used. Default value of register-suppression time is 60 seconds.
Command Mode	Global Configuration
Syntax	ip pim register-suppression <1-65535> ip pim (vrf NAME|) register-suppression <1-65535> no ip pim register-suppression no ip pim (vrf NAME|) register-suppression
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **<1-65535>**: Register suppression time in seconds
Example usage	`switch_a(config)# ip pim register-suppression 180`

ip pim rp-candidate	
Purpose	Give the router a candidate RP status using the IP address of the specified interface.
Command Mode	Global Configuration
Syntax	ip pim rp-candidate IFNAME ip pim (vrf NAME |) rp-candidate IFNAME no ip pim rp-candidate (IFNAME |) no ip pim (vrf NAME |) rp-candidate (IFNAME |)
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **IFNAME**: Interface name
Example usage	`switch_a(config)# ip pim rp-candidate ge10`

ip pim rp-address	
Purpose	Configure static RP address for multicast groups. PIM supports multiple static RPs. It also supports usage of static-RP and BSR mechanism, simultaneously. • If RP-address configured through BSR and RP-address configured statically are both available for a group range, the RP-address configured through BSR is chosen. • One static-RP can be configured for multiple group ranges using Access Lists. However, configuring multiple static RPs (using ip pim rp-address command) with the same RP address is not allowed. The static-RP can either be configured for the whole multicast group range 224/4 (without ACL) or for specific group ranges (using ACL). • If multiple static-RPs are available for a group range, then one with the highest IP address is chosen. • Only Permit filters in ACL are considered as valid group ranges. The default Permit filter 0.0.0.0/0 is converted to default multicast filter 224/4. • After configuration, the RP-address is inserted into static-RP group tree based on the configured group ranges. For each group range multiple static-RPs are maintained in a linked list, sorted by IP addresses. When selecting static-RPs for a group range, the first element, which is the static-RP with highest IP address, is chosen. • Deletion of RP-address is handled by removing the static-RP from all the existing group ranges and recomputing the RPs for existing TIB states if required. • Group mode and RP address mappings learned through BSR take precedence over mappings statistically defined by the ip pim rp-address command without the override keyword. Commands with the override keyword take precedence over dynamically learned mappings.
Command Mode	Global Configuration
Syntax	[no] ip pim rp-address A.B.C.D (override |) [no] ip pim rp-address A.B.C.D (<1-99> | <1300-1999> | WORD) (override |) [no] ip pim (vrf NAME |) rp-address A.B.C.D (override |) [no] ip pim (vrf NAME |) rp-address A.B.C.D (<1-99> | <1300-1999> | WORD) (override |)
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **<1-99>**: IP Standard access-list **<1300-1999>**: IP Standard access-list (expanded range) **WORD**: Access-list name **override**: Static RP overrides dynamically-learned RP
Example usage	`switch_a(config)# ip pim rp-address 3.3.3.3 4`

ip pim rp-register-kat	
Purpose	Configure a Keepalive Timer (KAT) value for (S,G) states at RP to monitor PIM register packets, overriding the generic KAT timer value.
Command Mode	Global Configuration
Syntax	ip pim rp-register-kat <1-65535> ip pim (vrf NAME |) rp-register-kat <1-65535> no ip pim rp-register-kat no ip pim (vrf NAME |) rp-register-kat
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **<1-65535>**: Keepalive timer in seconds
Example usage	`switch_a(config)# ip pim rp-register-kat 3454`

ip pim spt-threshold	
Purpose	Turn on the ability of the last-hop PIM router to switch to shortest-path tree (SPT). This option is binary, meaning that the switching to SPT happens either at the receiving of the first data packet or not at all. It is not rate-based.
Command Mode	Global Configuration
Syntax	ip pim spt-threshold ip pim spt-threshold group-list (<1-99> | <1300-1999> | WORD) ip pim (vrf NAME |) spt-threshold ip pim (vrf NAME |) spt-threshold group-list (<1-99> | <1300-1999> | WORD) no ip pim (vrf NAME |) spt-threshold no ip pim (vrf NAME |) spt-threshold group-list (<1-99> | <1300-1999> | WORD)
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **group-list**: Enable the ability for the last-hop PIM router to switch to SPT for multicast group addresses indicated by the given access-list **<1-99>**: IP Standard access-list **<1300-1999>**: IP Standard access-list (expanded range) **WORD**: Standard access list name
Example usage	`switch_a(config)# ip pim spt-threshold group-list LIST1`

ip pim ssm	
Purpose	Configure Source Specific Multicast (SSM), and define the range of multicast addresses. To define the SSM range to be other than the default, define an access-list. When an SSM range of IP multicast addresses is defined with the **ip pim ssm** command, the no (*,G) or (S,G,rpt) state is initiated for groups in the SSM range. The messages corresponding to these states are no accepted or originated in the SSM range.
Command Mode	Global Configuration
Syntax	ip pim ssm default ip pim ssm range (<1-99> | WORD) ip pim (vrf NAME |) ssm default ip pim (vrf NAME |) ssm range (<1-99> | WORD) no ip pim ssm no ip pim (vrf NAME |) ssm
Parameters	**vrf**: VPN routing/forwarding instance **NAME**: Name of the VPN routing/forwarding instance **default**: This keyword defines the 232/8 group range for SSM **range**: Define an access-list for group range to use for SSM **<1-99>**: Value for a standard access-list **WORD**: Standard access list name
Example usage	`switch_a(config)# ip pim ssm range 4`

PIM Interface Commands

ip pim bsr-border	
Purpose	Prevent bootstrap router (BSR) messages from being sent or received through an interface. Use this command to configure an interface bordering another PIM domain to avoid the exchange of BSR messages between the two domains. This prevents routers in one domain from electing rendezvous points (RPs) in the other domain, resulting in a protocol malfunction or loss of isolation between the domains.
Command Mode	Interface Configuration
Syntax	[no] ip pim bsr-border
Parameters	None
Example usage	`switch_a(config-if)# ip pim bsr-border`
Note	This command does not set up multicast boundaries. It only sets up a PIM domain BSR message border.

ip pim	
Purpose	Enable PIM dense-mode or sparse-mode on the current interface.
Command Mode	Interface Configuration
Syntax	[no] ip pim (dense-mode | sparse-mode)
Parameters	None
Example usage	`switch_a(config-if)# ip pim dense-mode`

ip pim dr-priority	
Purpose	Set the designated router's priority value.
Command Mode	Interface Configuration
Syntax	[no] ip pim dr-priority (<0-4294967294> |)
Parameters	**<0-4294967294>**: Designated router priority. A higher value means a higher preference.
Example usage	`switch_a(config-if)# ip pim dr-priority 314159`

ip pim exclude-genid	
Purpose	Exclude the GenID (generated ID) option from Hello packets sent by the PIM module on an interface. This command is used to inter-operate with older Cisco IOS versions.
Command Mode	Interface Configuration
Syntax	[no] ip pim exclude-genid
Parameters	None
Example usage	`switch_a(config-if)# ip pim exclude-genid`

ip pim hello-holdtime	
Purpose	Configure a hello holdtime other than the default (3.5 * hello_interval seconds). If the configured value is less than the current **hello_interval**, it is refused. When removing a configured hello_holdtime, the value is reset to default. Every time the **hello_interval** is updated, the **hello-holdtime** is checked. If it is less than the current **hello_interval** value, then it reverts to default. Otherwise, the configured value is maintained.
Command Mode	Interface Configuration
Syntax	ip pim hello-holdtime <1-65535> no ip pim hello-holdtime
Parameters	**<1-65535>:** Hello holdtime in seconds
Example usage	`switch_a(config-if)# ip pim hello-holdtime 20000`

ip pim hello-interval	
Purpose	Configure a hello interval value other than the default. When a hello-interval is configured and **hello-holdtime** is not configured, or when the **hello-holdtime** value configured is less than the new hello-interval value, the holdtime value is modified to (3.5 * hello_interval). Otherwise, the **hello-holdtime** value is the configured value.
Command Mode	Interface Configuration
Syntax	ip pim hello-interval <1-65535> no ip pim hello-interval
Parameters	**<1-65535>**: Hello interval in seconds. Default is 30 seconds.
Example usage	`switch_a(config-if)# ip pim hello-interval 300`

ip pim neighbor-filter	
Purpose	Enable filtering of neighbors on the interface. When configuring a neighbor filter, PIM will either not establish adjacency with neighbor or terminates adjacency with existing neighbors, when denied by filtering access list.
Command Mode	Interface Configuration
Syntax	[no] ip pim neighbor-filter (<1-99> | WORD)
Parameters	**<1-99>**: IP standard access-list number **WORD**: IP standard access list name
Example usage	`switch_a(config-if)# ip pim neighbor-filter 14`

ip pim propagation-delay	
Purpose	Configure a propagation delay value for PIM in mlliseconds.
Command Mode	Interface Configuration
Syntax	ip pim propagation-delay <1000-5000> no ip pim propagation-delay
Parameters	**<1000-5000>**: Propogation delay in milliseconds. Default is 500.
Example usage	`switch_a(config-if)# pim propagation-delay 1000`

ip pim state-refresh origination-interval	
Purpose	Configure a PIM-DM State-Refresh origination interval. This is the number of seconds between PIM-DM State Refresh control messages.
Command Mode	Interface Configuration
Syntax	ip pim state-refresh origination-interval <1-100> no ip pim state-refresh origination-interval
Parameters	**<1-100>**: Interval in seconds. Default is 60 seconds.
Example usage	`switch_a(config-if)# ip pim state-refresh origination-interval 72`

ip pim unicast-bsm	
Purpose	Enable support for sending and receiving unicast Bootstrap Messages (BSM) on an interface. This command supports backward-compatibility with older versions of the Bootstrap Router specification, which specifies unicast BSM to refresh the state of new or restarting neighbors.
Command Mode	Interface Configuration
Syntax	[no] ip pim unicast-bsm
Parameters	None
Example usage	`switch_a(config-if)# ip pim unicast-bsm`

24 Index of Commands

All CLI commands in alphabetical order.

E

F

H

I

L

M

N

O

P

R

S

T

25 Contact Information

EtherWAN System, Inc.

www.etherwan.com

USA Office

2301 E. Winston Road
Anaheim, CA 9280
Tel: +1-714-779-3800
Email: info@etherwan.com

Pacific Rim Office

8F., No.2, Alley 6, Lane 235, Baoqiao Rd.
Xindian District, New Taipei City 231
Taiwan
Tel: +886 -2- 6629-8986
Email: info@etherwan.com.tw

EG99000 Layer 3 Hardened Managed Ethernet Switch
December 11, 2018

Document version: Version 1