

Cisco Embedded Service 6300 Series Software Configuration Guide

First Published: 2019-03-28

Last Modified: 2026-09-28

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

© 2019–2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1 **Overview** 1

ESR6300 embedded router modules 1

Access the CLI using a router console on the ESR6300 router 2

Use the console interface 3

Remote console access 4

Telnet router console connection preparation 4

Use Telnet to access a console interface 5

CLI session management 6

Change the CLI session timeout 6

Lock a CLI session 7

CHAPTER 2 **New Features** 9

New features for Cisco IOS XE 26.2.1 9

New features for Cisco IOS XE 17.14.1a 9

New features for Cisco IOS XE 17.13.1 10

New features for Cisco IOS XE 17.12.1a 10

New features for Cisco IOS XE 17.11.1a 10

New features for Cisco IOS XE 17.10.1a 10

New features for Cisco IOS XE 17.9.1 10

New features for Cisco IOS-XE 17.8.1 11

Cellular serviceability enhancements 11

GNMI Broker update 12

gRPC Network Operations Interface updates 12

Raw socket feature enhancement 12

SCADA enhancement for TNB 12

DLEP and credit based radio aware routing support 13

New features for Cisco IOS XE 17.7.1 14
LTE support for ESR6300 14
Support 1G SFPs 14
New features for Cisco IOS-XE 17.6.1 15
Small Form Factor transceivers 15
VXLAN support 15
New features for Cisco IOS-XE 17.5.1 15
RFC4884 ICMPv6 and MPLSv6 15
Command example 16
Limitations 16
Netboot support 16
Alarm port support on the ESR6300 16
Alarm port configuration examples 18
New features for Cisco IOS-XE 17.4.1 21
Plug and Play (PnP) support 21
PnP overview 21
New features for Cisco IOS-XE 17.3.1 22
Security-Enhanced Linux (SELinux) support 22
Command examples 23
Syslog message reference 24
SD-WAN on the ESR6300 25
New features for Cisco IOS-XE 17.2.1 25
Serial port support 25
Boot from the USB 26
Boot from USB flash drive 26
Boot from ROMMON 27
Boot from USB feature summary 27

CHAPTER 3 **Web User Interface** 29
Web user interface management 29
Use Web User Interface for Day One setup 29
Configure LAN settings 30
Configure primary WAN settings 31
Configure secondary WAN settings 33

Configure security settings 34
GPS mode default enablement 34
Configure access point name using Cisco WebUI 35
Review and save router settings 38

CHAPTER 4 **Secure Shell 41**
Secure shell 41
Prerequisites for configure secure shell 41
Restrictions for configuring secure shell 41
SSH and router access 42
SSH servers, integrated clients, and supported versions 42
Recommendation: SSH configuration guidelines 43
How to configure secure shell 43
Set up the router to run SSH 44
Configure the SSH server 45
Monitor the SSH configuration and status 46
Configure the router for local authentication and authorization 47
Secure copy protocol 48
Requirements for secure copy configuration 48
Restrictions for configuring secure copy 49
Configure secure copy 49
SSH rekeying 50
SSH rekeying types 51
Prerequisites for configuring SSH rekeying 51
Restrictions for SSH rekeying on Cisco routers 52
Configure SSH rekeying 53
Verify SSH rekey configuration 54
Additional references 55

CHAPTER 5 **Using Cisco IOS XE Software 57**
Command modes 57
Use keyboard shortcuts 59
No and default command forms 59
Use the history buffer to recall commands 60

Configuration files management 60
Save configuration changes 61
Filtering output from show and more commands 61
Cisco feature navigator 62
Find support information for platforms and Cisco software images 62
Get help 63
Find command options 63
Software advisor 67
Use software release notes 67

CHAPTER 6 **Basic Router Configuration** 69
ESR6300 interface naming 69
Basic Configuration 70
Configuring Global Parameters 75
Serial port support 76
Rawsocket Keepalive Configuration CLI 76
Configuring the Gigabit Ethernet Interface 77
Configuring a Loopback Interface 78
Enabling Cisco Discovery Protocol 79
Configuring Command-Line Access 80
Configuring Static Routes 81
Configuring Dynamic Routes 83
Configuring Routing Information Protocol 83
Configuring Enhanced Interior Gateway Routing Protocol 84
Configuring Open Shortest Path First (OSPF) 84
Configuring Integrated Intermediate System-to-Intermediate System (IS-IS) Routing Protocol 85
Modular QoS (MQC) 85

CHAPTER 7 **Cisco IOS XE installation methods** 87
Bundle mode versus install mode 87
Installing software using install commands 88
Restrictions for installing the software using install commands 88
Install mode support 88
Information about installing the software using install commands 89

Installing software using install mode commands 90
Boot the platform in install mode 91
Convert from bundle mode to install mode or upgrade the platform using one-step installation 92
Three-step installation 93
Upgrading in install mode 95
Downgrading in install mode 95
Software installation termination 96
Configuration examples 96
One step installation 96
Complete three-step software installation 98
Show the installed packages 101
Show committed and uncommitted packages 102
Remove inactive packages 102
Troubleshoot software installation using install commands 103

CHAPTER 8 **DHCP Option 82** 105
DHCP Option 82 105
Prerequisites 106
Key considerations 107
Limitations 107
DHCP Option 82 packet flow 107
Hexadecimal pattern and suboption decoding 108
Configure DHCP Option 82 on server 109
Configure DHCP address pool 111
Configure DHCP relay agent 112
Configure relay chaining on relay agents 113
Verify DHCP Option 82 114
Debug DHCP Option 82 logs 115

CHAPTER 9 **IKEv2 and IPsec PFS Post Quantum Cryptography** 117
IKEv2 and IPsec PFS Post Quantum Cryptography 117
Benefits of PQC 118
Prerequisites 118
Limitations 118

Phased migration in Hub-and-Spoke deployments 119
ML-KEM algorithm and parameter sets 119
PFS negotiation behavior 120
Configure an IKEv2 proposal for PQC 120
Configure Perfect Forward Secrecy with PQC 122
Verify the PQC hybrid key exchange 123
Troubleshooting PQC hybrid key exchange 124

CHAPTER 10 **Segment Routing over MPLS** 125
Segment Routing over MPLS 125
Prerequisites 126
Configure Segment Routing over MPLS 126
Configure Segment Routing MPLS on OSPF network 127
Configure Prefix-SID for OSPF 128
Verify SR-MPLS configuration 129

CHAPTER 11 **Post Quantum Cryptography for SSHv2 sessions** 131
Implementation of Post Quantum Cryptography 131
Overview 132
Factors to consider for deploying PQC 132
Protocol requirements 132
Restrictions for PQC 132
PQC negotiation scenarios 133
Configure PQC for SSHv2 sessions 133
Verify PQC for SSHv2 sessions 134
Key elements of PQC 134
Supported platforms 136

CHAPTER 12 **Installing the Software** 137
Overview 137
Licensing 137
Cisco software licensing 138
Configure consolidated packages 138
License installation on the ESR6300 139

Network-Essentials 139
Network-Advantage 140
Policy: feature limitations for ESR6300 device deployment 141
Install the software for Cisco IOS XE 141
Boot from USB feature summary 143
Boot from ROMMON 143
Cisco IOS XE release upgrade 144
IOS XE downgrade warning 145
ROMMON images 146
File systems 146
Auto-generated file directories and files 147
Flash storage 148
LED indicators 148

CHAPTER 13 Smart Licensing Using Policy 149
Smart licensing using policy on Cisco routers 149

CHAPTER 14 Configuring Ethernet Switch Ports 151
Configuring VLANs 151
VLAN trunking protocol (VTP) 153
802.1x authentication 153
Configure spanning tree protocol 154
MAC address table manipulation 157
Configure switch port analyzer 158
IGMP snooping 159

CHAPTER 15 Gigabit Ethernet Combo Ports WAN 161
Gigabit ethernet combo ports WAN 161
Configure the WAN port media type 162

CHAPTER 16 SD-WAN Remote Access 165
SD-WAN 165

CHAPTER 17 CLI Output for the FN980 5G Modem 167
Change in CLI output for the FN980 5G modem 167

CHAPTER 18 MAB 802-1x Support 169
MAB 802.1x support 169

CHAPTER 19 Enable Secure Data Wipe Capabilities 171
Enable the secure data wipe capability 171

CHAPTER 20 Implementing Multicast Service Reflection 173
Multicast service reflection 173
Prerequisites for implementing multicast service reflection 173
Restrictions for implementing multicast service reflection 173
Information about implementing multicast service reflection 174
Benefits of implementing multicast service reflection 174
Rendezvous points 175
Vif1 interface 176
Multicast service reflection application 176
How to implement multicast service reflection 177
Configure multicast service reflection 177
Configuration examples for multicast service reflection 178
Example multicast-to-multicast destination translation 178
Example multicast-to-unicast destination translation 180
Configure multicast-to-multicast destination splitting 182
Example multicast-to-unicast destination splitting 184
Configure PIMv6 Anycast RP solution 186

CHAPTER 21 Cellular Boot Time Improvements 189
Cellular boot time improvements 189

CHAPTER 22 Cellular Pluggable Interface Modules 191

CHAPTER 23 **Radio Aware Routing and Dynamic Link Exchange Protocol** 199

Radio aware routing 200

Benefits of radio aware routing 202

Restrictions for radio aware routing and DLEP features 202

Performance 202

System components 203

Radio aware routing topology 204

QoS provisioning on PPPoE extension session 207

Configure the RAR feature in bypass mode 207

Verifying RAR Session Details 209

DLEP 214

DLEP topology and packet flow 214

License requirements 216

Configuring DLEP with OSPFv3 216

Configure the virtual multipoint interface 217

Configure the virtual template 218

Configure the physical interface 219

DLEP configuration modes 220

Configure the DLEP client and server based on port number 220

Configure DLEP with dynamic port on server (router) 222

Attach the DLEP template in discovery mode 224

Use a DLEP template with a well-known IP address 226

DLEP configuration with GTSM 227

Configure DLEP with OSPFv3 228

Configure DLEP with EIGRP 230

DLEP with quality of service (QoS) 231

Edit the virtual template 235

Configure DLEP on a sub-interface 236

Remove the DLEP configuration 239

Configure DLEP using the WebUI 240

DLEP validation commands 241

Optional configurations for DLEP 244

Restrictions for DLEP IPv6 unicast 244

IPv4 and IPv6 multicast over DLEP 245
IPv6 control plane for DLEP 246
DLEP client and neighbor clearing 247
Troubleshoot with show commands 248
Troubleshoot with debug commands 250
Additional debug commands 252
Related documentation 253

CHAPTER 24 **IPv6 Multicast Over PPPoE** 255
PPPoE in MANET 255
Requirement: feature requirements for IPV6 multicast over PPPoE 258
Enable IPv6 routing 258
Create a subscriber profile 259
Configure the license level 260
Configure PPPoE service policy 260
Configure QoS provisioning 261
Configure PPPoE service selection 262
Configure PPPoE on an ethernet interface 263
Configure a virtual template interface 264
Configure VMI interface and enable multicast support 266
Configure the loopback interface 267
Configure the OSPFv3 IPv4 address family process 268
Configure the OSPFv3 IPv6 address family process 270
Configure the IPv6 PIM bootstrap router 271
Configure an IPv6 multicast group 271
Verify virtual template interface 272
Verify PPPoE session details 273
Verify BSR election 274
IPv6 multicast routing table 275
IPv6 Multicast over PPPoE 276
Verify VMI neighbors 276
Verify OSPF neighbor 278
Sample Running Configuration 278
Debug commands 283

CHAPTER 25 **Implementation Options** 285

SFP Overview 285

SFP Command Line 286

Power over Ethernet 286

Device Detection and Power Allocation 287

CLI for PoE 287

Working with External USB3.0 289

CHAPTER 26 **ROM Monitor Overview and Basic Procedures** 291

ROM Monitor Overview 291

Access ROM Monitor Mode 292

Checking the Current ROMMON Version 292

Commonly Used ROM Monitor Commands 293

ROMMON Command Examples 294

Changing the ROM Monitor Prompt 294

Displaying the Configuration Register Setting 294

Environment Variable Settings 295

Frequently Used Environmental Variables 295

Displaying Environment Variable Settings 295

Entering Environment Variable Settings 295

Saving Environment Variable Settings 296

Exiting ROM Monitor Mode 296

Configuration Example 296

Upgrading the ROMMON for a Router 297

CHAPTER 27 **Device Zeroization** 299

Push Button 299

Microcontroller Unit (MCU) 300

Device Zeroization 300

Zeroization Trigger 301

Command Line Interface 302

Zeroization Support in bootloader 303

Troubleshooting Zeroization 303

CHAPTER 28 **Call Manager Express** 305

Call Manager Express Overview 305

SCCP Configuration Example 305

SCCP Show Command Example 306

SIP Configuration Example 307

Debug Commands 308

SIP Show Command Example 308

CHAPTER 29 **Troubleshooting** 311

Getting Started with Troubleshooting 311

Understanding Diagnostic Mode 311

Before Contacting Cisco or Your Reseller 312

show interfaces Troubleshooting Command 312

Software Upgrade Methods 312

Recovering a Lost Password 313

Change the Configuration Register 313

Configuring the Configuration Register for Autoboot 314

Reset the Router 314

Reset the Password and Save Your Changes 316

Reset the Configuration Register Value 316

Configuring a Console Port Transport Map 317

Viewing Console Port, SSH, and Telnet Handling Configurations 319

Using the factory reset Commands 320

CHAPTER 30 **System Messages** 321

Information About Process Management 321

How to Find Error Message Details 321

CHAPTER 31 **Process Health Monitoring** 327

Monitoring Control Plane Resources 327

Avoiding Problems Through Regular Monitoring 327

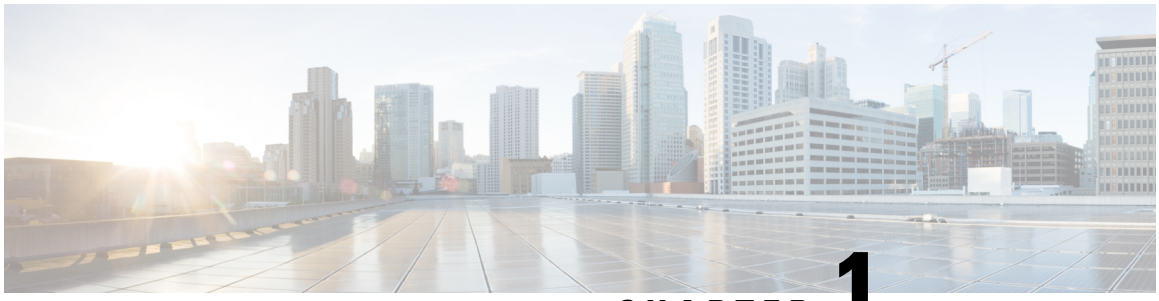
Cisco IOS XE Process Resources 327

Overall Control Plane Resources 331

Monitoring Hardware Using Alarms 332
Router Design and Monitoring Hardware 333
BootFlash Disk Monitoring 333
Approaches for Monitoring Hardware Alarms 333
Viewing the Console or Syslog for Alarm Messages 333
Enabling the logging alarm Command 333
Network Management System Alerts a Network Administrator when an Alarm is Reported Through SNMP 333

CHAPTER 32 **Environmental Monitoring** 335
Environmental Monitoring 335
Environmental Monitoring 335
Environmental Monitoring and Reporting Functions 336
Environmental Monitoring Functions 336
Environmental Reporting Functions 337
SNMP Polling of Temperature OID 343
Technical Assistance 344
Additional References 344

CHAPTER 33 **Additional Information** 345
Where to find Additional Information 345
Communications, Services, and Additional Information 345
Cisco Bug Search Tool 345
Documentation Feedback 346



CHAPTER 1

Overview

This section contains the following topics:

- ESR6300 embedded router modules, on page 1
- Access the CLI using a router console on the ESR6300 router, on page 2
- Remote console access, on page 4
- CLI session management, on page 6

ESR6300 embedded router modules

An ESR6300 is a compact form factor embedded router module that:

- provides a board size of 3.0" x 3.75" for integration into PC/104-compatible enclosures with adaptation,
- simplifies integration and offers system integrators flexibility for use in a wide variety of embedded applications, and
- runs IOS-XE, a Linux-based OS with enhanced features compared to classic IOS.

ESR6300 configuration options

The ESR card is available in two configuration options:

- ESR-6300-CON-K9: With a Cisco-designed cooling plate customized to the ESR.
- ESR-6300-NCP-K9: Without the cooling plate for system integrators who want to design their own custom thermal solution.

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

Note IOx development is not supported on the ESR6300. While this is platform independent code, it is unsupported and untested on this device.

Complete details on the ESR6300 are found in the product data sheet that is located here:

https://www.cisco.com/c/dam/en/us/products/routers/cisco-embedded-series-router-c78-742901.pdf

Access the CLI using a router console on the ESR6300 router

Procedure

Step 1 Connect your PC or laptop to the ESR6300 router console port using a USB or RS232 cable.

Note
If your computer warns that the proper drivers are missing, obtain them from your computer manufacturer or from the USB-to-UART bridge driver site.

Step 2 Power on the router. If the device is fresh from the factory, the System Configuration Dialog appears.

Example:

```
--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: yes
```

Note
If the router is provisioned for Cisco PnP connect services, the initial dialog is skipped.

Step 3 Respond to the prompts in the System Configuration Dialog to perform basic management setup, including hostname and passwords.

Example:

```
Would you like to enter basic management setup? [yes/no]: yes

Enter host name [Router]: <your-host-name>

Enter enable secret: <your-password>

Enter enable password: <your-password>

Enter virtual terminal password: <your-password>
```

Step 4 Set up the HTTP server account and SNMP configuration as prompted.

Example:

```
Setup account for accessing HTTP server? [yes]: <return>
Username  [admin]: <your-username>
```

```
Password  [cisco]: <your-password>
Password is UNENCRYPTED.
Configure SNMP Network Management? [no]: <return>
```

Step 5 Review the interface summary and select the management interface to configure IP addressing.

Example:

```
            Enter interface name used to connect to the management network from the above
interface summary: vlan1

            Configure IP on this interface? [no]: yes
            IP address for this interface: 192.168.1.1
            Subnet mask for this interface [255.255.255.0]: <return>
```

Step 6 Configure DHCP settings for the network as prompted.

Example:

```
            Would you like to configure DHCP? [yes/no]: yes
            Enter DHCP pool name: wDHCPool
            Enter DHCP network: 192.168.1.0
            Enter DHCP netmask: 255.255.255.0
            Enter Default router: 192.168.1.1
```

Step 7 Review the generated configuration script and save the configuration to NVRAM.

Example:

```
            Enter your selection [2]: 2
            Press RETURN to complete the setup and access the router prompt.
```

Step 8 After setup completes, press RETURN to access the router CLI prompt and begin further configuration.

Use the console interface

Use this task to access the CLI through the console interface in privileged EXEC mode for performing administrative tasks on your router.

Procedure

Step 1 Use the **enable** command from user EXEC mode to enter privileged EXEC mode.

Example:

```
Router> enable
```

Step 2 Enter your system password at the password prompt. Skip this step if the enable password has not been configured. When your password is accepted, the privileged EXEC mode prompt is displayed.

Example:

```
Password: enablepass
            Router#
```

Step 3 Use the **quit** command to exit the console session.

Example:

```
Router# quit
```

Remote console access

Remote console access is a network connectivity method that:

- enables access to the ESR6300 CLI from a remote location,
- supports both Telnet and SSH protocols for connection establishment, and
- provides secure management capabilities through encrypted SSH connections.

Remote console access methods

The ESR6300 supports two primary methods for remote console access:

- **Telnet**: Standard protocol for remote terminal access.
- **SSH**: Secure Shell protocol providing encrypted communication.

For SSH configuration details, see Secure Shell, on page 41.

Telnet router console connection preparation

Telnet router console connection preparation is a configuration process that:

- configures virtual terminal lines to support remote access via Telnet,
- establishes authentication and password requirements for user login, and
- ensures proper hostname or IP address configuration for network connectivity.

Configuration requirements

To access the router remotely using Telnet from a TCP/IP network, configure the router to support virtual terminal lines using the **line vty** global configuration command. Configure the virtual terminal lines to require users to log in and specify a password.

To prevent disabling login on a line, specify a password with the **password** command when you configure the **login** command.

If you are using authentication, authorization, and accounting (AAA), configure the **login authentication** command. To prevent disabling login on a line for AAA authentication when you configure a list with the login authentication command, you must also configure that list using the **aaa authentication login** global configuration command.

In addition, before you make a Telnet connection to the router, you must have a valid hostname for the router or have an IP address configured on the router.

Configuring the diagnostic and wait banners is optional, but recommended. The banners are especially useful as indicators to users about the status of their Telnet or SSH attempts.

Note See the Cisco IOS-XE Device hardening guide at https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html for details.

Use Telnet to access a console interface

Use this task to access a router's console interface using Telnet to perform CLI management and configuration tasks.

Note If you are using an access server, specify a valid port number, such as telnet 172.20.52.40 2004, in addition to the hostname or IP address.

Procedure

Step 1 Use the **connect** *host* [*port*] [*keyword*] or **telnet** *host* [*port*] [*keyword*] command from your terminal or PC to connect to the router. Here, *host* is the router hostname or IP address, *port* is a decimal port number (23 is the default), and *keyword* is a supported keyword.

Example:

```
unix_host% telnet router
           Trying 172.20.52.40...
           Connected to 172.20.52.40.
           Escape character is '^]'.
           unix_host% connect
```

Step 2 Enter your login password at the prompt. If no password has been configured, press Return.

Example:

```
User Access Verification
           Password: mypassword
```

Step 3 Use the **enable** command from user EXEC mode to enter privileged EXEC mode.

Example:

```
Router> enable
```

Step 4 Enter your system password at the password prompt. When the enable password is accepted, the privileged EXEC mode prompt is displayed.

Example:

```
Password: enablepass
           Router#
```

Step 5 Use the **exit** or **logout** command to exit the Telnet session.

Example:

```
Router# logout
```

CLI session management

CLI session management is a network device feature that:

- enforces configurable inactivity timeouts to automatically terminate idle sessions,
- provides session locking protection to prevent multiple users from overwriting each other's changes, and
- reserves spare capacity for CLI session access to ensure remote access availability even when internal processes consume system resources.

Change the CLI session timeout

Use this task for setting the CLI session timeout. It increases the security of a CLI session by automatically ending idle sessions after a specified period.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **line console** *console line -number* command to access the console line configuration mode.

Example:

```
Router(config)# line console 0
```

Step 3 Use the **session-timeout** *minutes* command to set the CLI session timeout interval.

Example:

```
Router(config-line)# session-timeout 10
```

Note
The *minutes* value sets the amount of time the CLI waits before timing out. Specify 0 to disable the session timeout.

Step 4 Use the **show line console** *console line -number* command to verify the session timeout setting, shown as the value for "Idle Session".

Example:

```
Router# show line console 0
```

Lock a CLI session

Use this task to lock a CLI session which prevents unauthorized access by requiring a temporary password to unlock the session.



Note Before you can use the lock command, you need to configure the line using the lockable command.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **line console** *line-number* command to enter the line upon which you want to be able to use the lock command.

Example:

```
Router(config)# line console 0
```

Step 3 Use the **lockable** command to enable the line to be locked.

Example:

```
Router(config)# lockable
```

Step 4 Use the **exit** command to exit line configuration mode.

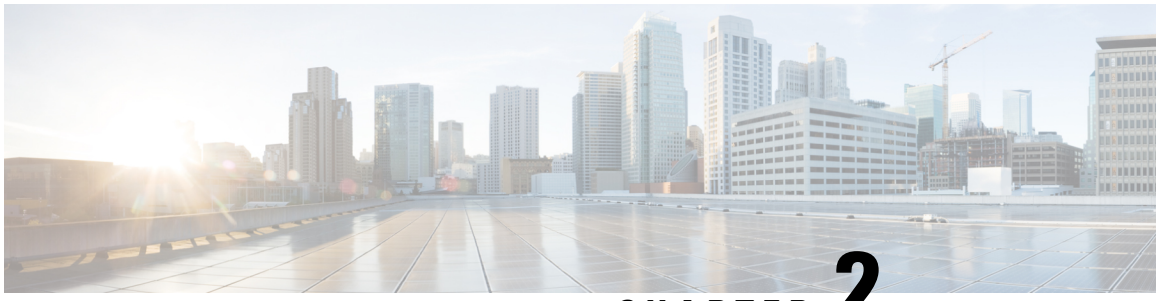
Example:

```
Router(config)# exit
```

Step 5 Use the **lock** command to lock the CLI session. The system prompts you for a password, which you must enter twice.

Example:

```
Router# lock
            Password: <password>
            Again: <password>
            Locked
```



CHAPTER 2

New Features

- New features for Cisco IOS XE 26.2.1, on page 9
- New features for Cisco IOS XE 17.14.1a, on page 9
- New features for Cisco IOS XE 17.13.1, on page 10
- New features for Cisco IOS XE 17.12.1a, on page 10
- New features for Cisco IOS XE 17.11.1a, on page 10
- New features for Cisco IOS XE 17.10.1a, on page 10
- New features for Cisco IOS XE 17.9.1, on page 10
- New features for Cisco IOS-XE 17.8.1, on page 11
- New features for Cisco IOS XE 17.7.1, on page 14
- New features for Cisco IOS-XE 17.6.1, on page 15
- New features for Cisco IOS-XE 17.5.1, on page 15
- New features for Cisco IOS-XE 17.4.1, on page 21
- New features for Cisco IOS-XE 17.3.1, on page 22
- New features for Cisco IOS-XE 17.2.1, on page 25

New features for Cisco IOS XE 26.2.1

New features in this release are listed below:

- IKEv2 and IPsec PFS Post Quantum Cryptography
- DHCP Option 82
- Post Quantum Cryptography for SSHv2 sessions
- Segment Routing over MPLS

New features for Cisco IOS XE 17.14.1a

New features in this release are listed below:

- SNMP MIB Support for DLEP

New features for Cisco IOS XE 17.13.1

This chapter contains the new features and enhancements available in Cisco IOS XE 17.13.1 release, organized by feature category.

- IPv4 and IPv6 Multicast Over DLEP
- IPv6 Control Plane for DLEP
- SD-WAN Remote Access (SD-WAN RA)
- Change in CLI Output for the FN980 5G Modem

New features for Cisco IOS XE 17.12.1a

This chapter contains these sections:

- DLEP IPv6 Unicast
- Uncapped License Implementation

New features for Cisco IOS XE 17.11.1a

This chapter contains information about the new features available in Cisco IOS XE 17.11.1a release, including changes to Smart Licensing Packaging and Galileo Support on the LTE Pluggable Modules.

- Change to Smart Licensing Packaging
- Galileo Support on the LTE Pluggable Modules

New features for Cisco IOS XE 17.10.1a

This reference provides information about the new features available in Cisco IOS XE 17.10.1a. The following sections detail each new feature and its implementation.

- Support for the P-5GS6-GL Pluggable Module on the ESR6300
- MAB 802.1x Support
- Enable Secure Data Wipe Capabilities
- Rawsocket Keepalive Configuration CLI

New features for Cisco IOS XE 17.9.1

This chapter contains the following sections:

- Install Mode Support
- Cellular Boot Time Improvements
- IOS XE Downgrade Warning
- SNMP Polling of Temperature OID
- GPS Mode Enabled By Default
- Cisco WebUI Access Point Name (APN)
- IPv6 Multicast over PPPoE

New features for Cisco IOS-XE 17.8.1

This reference provides information about new features introduced in Cisco IOS-XE 17.8.1.

Cellular serviceability enhancements

Cellular serviceability enhancements are debugging features that:

- enable trigger points and debug code via controller cellular CLIs for automatic generation and trapping of debug data without manual intervention,
- provide context-based debug logs including tracebacks and GPS locations, and
- create well-formatted debug messages with vendor-specific debug data at a broader range.

Available CLI options

The following CLI options are available for cellular serviceability:

```
(config-controller)#lte modem serviceability ?
  gps               GPS debugging
  interface-resets  Interface resets/Bearer deletion
  modem-crash       Modem-crash debugging
  modem-resets      IOS initiated unknown modem-resets
```

The debug data includes:

- Context Based debug logs (tracebacks, and GPS locations).
- Well formatted debug messages.
- Vendor specific debug data at a broader range.

The debug logs are located in the following location of flash:

```
router#dir flash:servelogs
Directory of bootflash:/servelogs/

259340  -rw-             122   Sep 7 2021 17:40:44 +00:00  gpslog-slot5-20210907-174044
259339  -rw-            1734   Sep 7 2021 12:14:07 +00:00  celllog-slot5-20210905-164628
```

GPS and cellular log files are created separately with file names using the timestamp at the time of the creation. These files are created as follows:

- If the existing file has reached 10Mb, a new file will be created.
- A new file will be created if the feature (GPS, or cellular) is completely disabled, and then re-enabled.

GNMI Broker update

A GNMI Broker update is a software enhancement that:

- extends GNMIB to support the gRPC Network Operations Interface (gNOI) reset.proto service,
- provides functionality for restoring the device to its factory defaults via gRPC, and,
- maintains the current booted image through the reset process.

Service behavior and functionality

When the service is executed, it behaves similarly to the 'factory-reset all' command, and subsequently triggering a reload. Additionally, the service will maintain the current booted image. These additional steps will be taken to comply with the reset.proto service:

- Set the rommon BOOT variable to the current booted image and maintain it through reload following factory-reset,
- Enable autoboot to bring the device up on the current booted image following factory-reset.

gRPC Network Operations Interface updates

A gRPC Network Operations Interface (gNOI) update is a gRPC-based microservice that:

- defines interfaces for executing operational commands and procedures on network devices, such as OS Install, Activate, and Verification,
- enables operating system related tasks through os.proto, including OS activation, install, detailed overview, internal OS commands, and summary output of OS operations, and
- provides capabilities to display gnmib detailed state, check gnmib operational statistics, and output modifiers.

Raw socket feature enhancement

A raw socket feature enhancement is a network configuration capability that:

- allows the user to input the maximum number of retries available to the write socket,
- provides a range from 1 to 1000 retries with a default of 10 retries, and
- includes a new CLI command **raw-socket tcp max-retries <1-1000>** where <1-1000> is the maximum number of retries.

SCADA enhancement for TNB

This enhancement provides compatibility with TNB's WG RTUs, including:

- TNB RTUs require Reset-Link message to be sent out along with Link-Status message to ensure correct initialization of the serial. The feature can be selectively turned on using the new configuration CLI **scada-gw protocol force reset-link.**
- When clock passthru is enabled and if the router hasn't received the timestamp from the DNP3-IP master, the router's hardware time will be sent downstream to RTU. Upon receiving a new timestamp from DNP3-IP master, the router will start sending the new timestamp sourced from DNP3-IP master to RTU.
- The number of bufferable DNP3 events in memory will be increased from 600 to 10000.
- The **scada-gw protocol interlock** command will be supported for DNP3. Previously, the support only existed for T101/T104. With this new enhancement, the router will disconnect Serial link if the DNP3-IP master is down or unreachable. Similarly, when the Serial link to RTU is down, the TCP connection to DNP3-IP master will be untethered.
- Custom "requests" will be automatically ordered based on priority so that the user can specify them in any order that they would like to.

DLEP and credit based radio aware routing support

DLEP and credit based radio aware routing support are networking technologies that:

- address challenges when merging IP routing and radio frequency (RF) communications,
- enable optimal route selection based on feedback from radios with faster convergence, and
- provide flow-controlled communications using rate-based Quality of Service (QoS) policies.

DLEP support

Cisco provides capabilities that enable:

- Optimal route selection based on feedback from radios,
- Faster convergence when nodes join and leave the network,
- Efficient integration of point-to-point, point-to-multipoint and broadcast multi-access radio topologies with multi-hop routing,
- Flow-controlled communications between the radio and its partner router using rate-based Quality of Service (QoS) policies,
- Dynamic shaping of fluctuating RF bandwidth in near real time to provide optimized use of actual RF bandwidth.

Radio-Aware Routing (RAR) is a mechanism that uses radios to interact with the routing protocol OSPFv3 to signal the appearance, disappearance, and link conditions of one-hop routing neighbors.

In a large mobile networks, connections to the routing neighbors are often interrupted due to distance and radio obstructions. When these signals do not reach the routing protocols, protocol timers are used to update the status of a neighbor. Routing protocols have lengthy timer, which is not recommended in mobile networks.

PPPoE extensions are used when the router communicates with the radio. In the Cisco IOS implementation of PPPoE, each individual session is represented by virtual access interface (connectivity to a radio neighbor) on which, QoS can be applied with these PPPoE extensions.

RFC5578 provides extensions to PPPoE to support credit-based flow control and session-based real time link metrics, which are very useful for connections with variable bandwidth and limited buffering capabilities (such as radio links).

Complete details can be found in Radio Aware Routing and Dynamic Link Exchange Protocol.

New features for Cisco IOS XE 17.7.1

This reference provides information about the new features and enhancements introduced in Cisco IOS XE 17.7.1 release.

LTE support for ESR6300

Cisco IOS-XE Release 17.7.1 adds support for LTE modules on the ESR6300 platform in slot 3. The related interface is Cellular 0/3/0.

Supported LTE modules

ESR6300 supports these LTE modules:

- P-LTE-MNA (WP7610 modem),
- P-LTEA-LA (EM7430 modem),
- P-LTEA-EA (EM7455 modem),
- P-LTEAP18-GL (LM960 modem).

The cellular interface/module configuration will follow the same as on any other Polaris IOS-XE based platform and LTE modem.

Support 1G SFPs

Release 17.7.1 will add support for these SFPs:

- GLC-T-RGD
- CWDM-SFP-1470=
- CWDM-SFP-1610=
- CWDM-SFP-1530=
- DWDM-SFP-3033=
- DWDM-SFP-3112=
- GLC-BX-D-I=
- GLC-BX-U-I=
- GLC-TE

New features for Cisco IOS-XE 17.6.1

This reference provides information about new features introduced in Cisco IOS-XE 17.6.1.

Small Form Factor transceivers

A Small Form Factor (SFF) transceiver is an optical transceiver that

- is soldered onto the PCB,
- tends to be smaller in size than traditional SFPs, and,
- operates in the same way as an SFP.

SFF support on ESR6300

The ESR6300 supports SFFs to meet high vibration and shock requirements that cannot be met with traditional SFPs. Support was added in 17.6.1 for the Finisar FTE8501K1LTN.

Since this is not a Cisco transceiver, the end-user will need to use the **service unsupported-transceiver** CLI in order to get the router to accept and initialize the SFF.

VXLAN support

This feature functions the same as it does on the IR1101 running 17.5.1

See the section on VXLAN in the IR1101 Software Configuration Guide.

New features for Cisco IOS-XE 17.5.1

This reference provides information about the new features and enhancements introduced in Cisco IOS-XE 17.5.1 release.

RFC4884 ICMPv6 and MPLSv6

RFC 4884 is a protocol enhancement that:

- redefines selected ICMP error messages to support multi-part operation,
- carries all of the information that ICMP messages carried previously, as well as additional information that applications may require, and
- introduces an 8-bit length attribute to ICMPv6 messages with extensions.

ICMPv6 message types and implementation details

The RFC 4884 feature introduces an 8-bit length attribute to these ICMPv6 messages with extensions:

- Destination Unreachable (type = 1)

- Time Exceeded (type = 3)

As part of RFC 4884 feature, for applications like MPLS/trace route which add extensions to type 1 and type 3 ICMPv6 error messages, original datagram length will be added in ICMPv6 header.

Also, infra is added as part of RFC 4884 support. If any new application is adding extensions it has to call defined registries to be compliant with RFC 4884.

Backward compatibility is also taken care of as part of this feature.

This feature is enabled by default and a CLI *[no]* **ipv6 icmp od-length enable** is provided which is enabled by default.

Command example

This reference provides a command example demonstrating the syntax for the **ipv6 icmp od-length enable** command.

```
ipv6 icmp od-length enable
```

Limitations

RFC4884 ICMP v4 and MPLS v4 extensions will be supported in the IOS-XE 17.6.1 release.

Netboot support

Netboot support is a recovery feature that:

- allows users to recover their router when there is no image in the bootflash or USB,
- uses TFTP boot functionality on the ESR6300 router, and
- operates through two Combo ports, Copper and Fiber ports (SFP) that support TFTP boot.

ROMMON configuration requirements

The following configuration needs to be in place in ROMMON:

- WAN port Gigabit-Ethernet 0/0/0 or 0/0/1 should be connected to a TFTP network.
- Path to image should be in a TFTP directory.
- set IP_ADDRESS=<IP address of router>.
- set IP_SUBNET_MASK=<mask>.
- set DEFAULT_GATEWAY=<IP address of gateway>.
- boot tftp://<server IP address>/<path to image>.

Alarm port support on the ESR6300

There is one alarm port available on the ESR6300. The IOS name for the alarm port is Alarm Contact 0.

The following configuration commands are available in IOS:

- alarm contact 0 enable
- alarm contact 0 description
- alarm contact 0 severity
- alarm contact 0 trigger

The configuration commands also have their equivalent **no** prefaces.

Alarm contact command

The ESR6300 supports only one alarm contact, which is Alarm Contact 0. Options are described in the following table:

description	The description string is up to 80 alphanumeric characters in length and is included in any generated system messages.
severity	For severity, enter critical, major, minor or none. If you do not configure a severity, the default is minor.
trigger	For trigger, enter open or closed. If you do not configure a trigger, the alarm is triggered when the circuit is closed.
enable	Provides a mechanism for you to enable or disable alarm conditions for a port.

Alarm trigger command

The **trigger** command has two options as shown:

closed	Assert alarm when contact is closed. Closed means that no current flows through the contact (normally open contact). The alarm is generated when current does flow.
open	Assert alarm when contact is open. Open means that the normal condition has current flowing through the contact (normally closed contact). The alarm is generated when the current stops flowing.

Note See the Alarm Port Configuration Examples for command examples.

Alarm LEDs

The alarm LED behavior is described in the table:

Severity	LED Status
Critical	Flashing Red
Major	Flashing Red
Minor	Red

Note The LED behavior depends on both the trigger configuration as well as the severity configuration. The LED behavior does not differentiate between the Critical and Major severity.

Alarm port configuration examples

To configure the feature, the alarm contact 0 needs to be enabled first.

Basic alarm contact configuration

```
Router#config term
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#alarm contact 0 enable
Router(config)#alarm contact 0 description test
Router(config)#alarm contact 0 severity critical
```

Alarm Enable/Disable

The alarm needs to be enabled to configure the severity and trigger. The following example shows the errors when the alarm is not enabled:

```
Router(config)#alarm contact 0 trigger open
Alarm / Digital IO Port 0 is not enabled.

Router(config)#alarm contact 0 severity major
Alarm / Digital IO Port 0 is not enabled.
```

Enabling the alarm and setting the severity

See the following example:

```
Router(config)#alarm contact 0 enable

Router(config)#alarm  contact 0 severity ?
  critical  Critical alarm severity
  major     Major alarm severity
  minor     Minor alarm severity
  none      No alarm severity

Router(config)#alarm contact 0 severity major
Router(config)#end
Router#
*Oct 16 14:54:54.518: %IIOT_ALARM_CONTACT-0-EXTERNAL_ALARM_CONTACT_ASSERT: External
alarm/digital IO port (External alarm contact on Motherboard) asserted

*Oct 16 14:54:54.518: %IIOT_ALARM_CONTACT-0-EXTERNAL_ALARM_CONTACT_ASSERT: External
alarm/digital IO port (ASSERT) asserted
```

```
ha
*Oct 16 14:54:54.733: %SYS-5-CONFIG_I: Configured from console by cons
```

Viewing the configuration

To view the configuration:

```
Router#show alarm
Alarm contact 0:
Description: External alarm contact on Motherboard
Status: Not Asserted
Application: Dry
Severity: major
Trigger: Open
Mode: Input
Router#

Router#show facility-alarm status
System Totals  Critical: 0  Major: 1  Minor: 0

Source                    Time                  Severity      Description [Index]
------                    ------                --------      -------------------

External alarm contact     Oct 16 2023 14:46:14   MAJOR         External alarm contact on
Motherboard [0]

Async0/2/0                Oct 15 2023 18:58:08   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/0/0      Oct 15 2023 18:58:21   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/0/1      Oct 15 2023 18:58:21   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/1/0      Oct 15 2023 18:58:20   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/1/1      Oct 15 2023 18:58:20   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/1/2      Oct 15 2023 18:58:20   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/1/3      Oct 15 2023 18:58:20   INFO          Physical Port Administrative
 State Down [2]

Cellular0/3/0             Oct 15 2023 18:58:22   INFO          Physical Port Administrative
 State Down [2]

Cellular0/3/1             Oct 15 2023 18:58:22   INFO          Physical Port Administrative
 State Down [2]

Router#
Router#show facility-alarm status major
System Totals  Critical: 0  Major: 1  Minor: 0

Source                    Time                  Severity      Description [Index]
------                    ------                --------      -------------------

External alarm contact     Oct 16 2023 14:46:14   MAJOR         External alarm contact on
Motherboard [0]
```

Alarm trigger commands

See the following example:

```
Router#show run | sec alarm
alarm contact 0 enable
alarm contact 0 trigger Open
logging alarm informational
Router#
Router#show alarm
Alarm contact 0:
   Description: External alarm contact on Motherboard
   Status:      Asserted
   Application: Dry
   Severity:    minor
   Trigger:     Open
   Mode:        Input
Router#
Router#show facility-alarm status
System Totals  Critical: 0  Major: 0  Minor: 1

Source                    Time                   Severity      Description [Index]
------                    ------                 --------      -------------------

External alarm contact    Oct 16 2023 14:54:54   MINOR         External alarm contact on
Motherboard [0]

Async0/2/0                Oct 15 2023 18:58:08   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/0/0      Oct 15 2023 18:58:21   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/0/1      Oct 15 2023 18:58:21   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/1/0      Oct 15 2023 18:58:20   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/1/1      Oct 15 2023 18:58:20   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/1/2      Oct 15 2023 18:58:20   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/1/3      Oct 15 2023 18:58:20   INFO          Physical Port Administrative
 State Down [2]

Cellular0/3/0             Oct 15 2023 18:58:22   INFO          Physical Port Administrative
 State Down [2]

Cellular0/3/1             Oct 15 2023 18:58:22   INFO          Physical Port Administrative
 State Down [2]

Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#alarm  contact 0 trigger closed
Router(config)#
*Oct 16 14:58:19.548: %IIOT_ALARM_CONTACT-0-EXTERNAL_ALARM_CONTACT_CLEAR: External
alarm/digital IO port (External alarm contact on Motherboard) cleared

*Oct 16 14:58:19.549: %IIOT_ALARM_CONTACT-0-EXTERNAL_ALARM_CONTACT_CLEAR: External
alarm/digital IO port (CLEAR) cleared
```

```
Router#sh facility-alarm status
System Totals  Critical: 0  Major: 0  Minor: 0

Source                    Time                   Severity      Description [Index]
------                    ------                 --------      -------------------

Async0/2/0                Oct 15 2023 18:58:08   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/0/0      Oct 15 2023 18:58:21   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/0/1      Oct 15 2023 18:58:21   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/1/0      Oct 15 2023 18:58:20   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/1/1      Oct 15 2023 18:58:20   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/1/2      Oct 15 2023 18:58:20   INFO          Physical Port Administrative
 State Down [2]

GigabitEthernet0/1/3      Oct 15 2023 18:58:20   INFO          Physical Port Administrative
 State Down [2]

Cellular0/3/0             Oct 15 2023 18:58:22   INFO          Physical Port Administrative
 State Down [2]

Cellular0/3/1             Oct 15 2023 18:58:22   INFO          Physical Port Administrative
 State Down [2]

Router#

Router#show alarm
Alarm contact 0:
   Description: External alarm contact on Motherboard
   Status:      Not Asserted
   Application: Dry
   Severity:    minor
   Trigger:     Closed
   Mode:        Input
```

New features for Cisco IOS-XE 17.4.1

This reference provides information about the new features and enhancements introduced in Cisco IOS-XE 17.4.1 release.

Plug and Play (PnP) support

This release enables PnP to work the same as on the IR1101. See the IR1101 Software Configuration Guide.

PnP overview

A PnP (Plug and Play) process is an automated device configuration mechanism that:

- provides zero-touch provisioning for network devices during initial boot-up

- eliminates manual configuration by automatically connecting to a PnP server and downloading device configurations, and
- enables remote configuration deployment without requiring console access to the router.

PnP process flow

The PnP discovery process follows this sequence:

1. The out of box configuration boots the platform up to the configuration wizard.
2. The control stops at a prompt where the user is given an option to enter the startup configuration wizard or not.
3. If the user does not have access to the router, or does not enter any options, PnP discovery kicks in.
4. If the PnP agent successfully establishes a connection to the PnP Server, the device configurations are pushed from the Server.
5. The platform gets configured according to the user preference.

If PnP is not setup for the Router, the WebUI is accessible without having to access the platform console.

New features for Cisco IOS-XE 17.3.1

This reference provides information about the new features introduced in Cisco IOS-XE 17.3.1 release.

Security-Enhanced Linux (SELinux) support

Security-Enhanced Linux is a set of patches to the Linux kernel and utilities that:

- incorporates a strong, flexible mandatory access control (MAC) architecture into the major subsystems of the kernel,
- provides an enhanced mechanism to enforce the separation of information based on confidentiality and integrity requirements, and
- enables the confinement of damage that can be caused by malicious or flawed applications.

Security enforcement and configuration details

SELinux enforces mandatory access control policies that confine user programs and system servers to the minimum amount of privilege they require to do their jobs. This reduces or eliminates the ability of these programs and daemons to cause harm when compromised (for example, via buffer overflows or mis-configurations). This confinement mechanism operates independently of the traditional Linux access control mechanisms.

The are no additional requirements or configuration steps required to enable or operate the SELinux feature. The solution is enabled/operational by default as part of the base IOS-XE software on supported platforms.

The following are enhanced show commands that have been defined for viewing SELinux related audit logs:

- **show platform software audit all**

- **show platform software audit summary**
- **show platform software audit switch** *<<1-8> | active | standby> <FRU identifier from a drop-down list>*

Command examples

This reference provides sample outputs for the **show software platform software audit summary** , **show software platform software audit all**, and **show software platform software audit switch** commands.

Sample output for show platform software audit summary command

```
Device# show platform software audit summary
====================================
AUDIT LOG ON switch 1
------------------------------------
AVC Denial count: 58
====================================
```

Sample output for show platform software audit all command

```
Device# show platform software audit all
====================================
AUDIT LOG ON switch 1
------------------------------------
========== START ============
type=AVC msg=audit(1539222292.584:100): avc:  denied  { read } for  pid=14017
comm="mcp_trace_filte" name="crashinfo" dev="rootfs" ino=13667
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=lnk_file permissive=1
type=AVC msg=audit(1539222292.584:100): avc:  denied  { getattr } for  pid=14017
comm="mcp_trace_filte" path="/mnt/sd1" dev="sda1" ino=2
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:101): avc:  denied  { getattr } for  pid=14028 comm="ls"
 path="/tmp/ufs/crashinfo" dev="tmpfs" ino=58407
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:102): avc:  denied  { read } for  pid=14028 comm="ls"
name="crashinfo" dev="tmpfs" ino=58407 scontext=system_u:system_r:polaris_trace_filter_t:s0
 tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539438600.896:119): avc:  denied  { execute } for  pid=8300 comm="sh"
name="id" dev="loop0" ino=6982 scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
 tcontext=system_u:object_r:bin_t:s0 tclass=file permissive=1
========== END ============
```

(output omitted for brevity)

Sample output for show platform software audit switch command

```
Device# show platform software audit switch active R0
========== START ============
type=AVC msg=audit(1539222292.584:100): avc:  denied  { read } for  pid=14017
comm="mcp_trace_filte" name="crashinfo" dev="rootfs" ino=13667
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=lnk_file permissive=1
type=AVC msg=audit(1539222292.584:100): avc:  denied  { getattr } for  pid=14017
comm="mcp_trace_filte" path="/mnt/sd1" dev="sda1" ino=2
```

```
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_disk_crashinfo_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:101): avc:  denied  { getattr } for  pid=14028 comm="ls"
 path="/tmp/ufs/crashinfo" dev="tmpfs" ino=58407
scontext=system_u:system_r:polaris_trace_filter_t:s0
tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539222292.586:102): avc:  denied  { read } for  pid=14028 comm="ls"
name="crashinfo" dev="tmpfs" ino=58407 scontext=system_u:system_r:polaris_trace_filter_t:s0
 tcontext=system_u:object_r:polaris_ncd_tmp_t:s0 tclass=dir permissive=1
type=AVC msg=audit(1539438624.916:122): avc:  denied  { execute_no_trans } for  pid=8600
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438648.936:123): avc:  denied  { execute_no_trans } for  pid=9307
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438678.649:124): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
type=AVC msg=audit(1539438696.969:125): avc:  denied  { execute_no_trans } for  pid=10057
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438732.973:126): avc:  denied  { execute_no_trans } for  pid=10858
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438778.008:127): avc:  denied  { execute_no_trans } for  pid=11579
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438800.156:128): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
type=AVC msg=audit(1539438834.099:129): avc:  denied  { execute_no_trans } for  pid=12451
comm="auto_upgrade_se" path="/bin/bash" dev="rootfs" ino=7276
scontext=system_u:system_r:polaris_auto_upgrade_server_rp_t:s0
tcontext=system_u:object_r:shell_exec_t:s0 tclass=file permissive=1
type=AVC msg=audit(1539438860.907:130): avc:  denied  { name_connect } for  pid=26421
comm="nginx" dest=8098 scontext=system_u:system_r:polaris_nginx_t:s0
tcontext=system_u:object_r:polaris_caf_api_port_t:s0 tclass=tcp_socket permissive=1
========== END ============
==================================
```

Syslog message reference

This reference provides syslog message details including the facility-severity-mnemonic format, severity level meanings, message explanations for resource access policy mismatches, and recommended troubleshooting actions.

Facility severity mnemonic

- %SELINUX-3-MISMATCH

Severity meaning

- ERROR LEVEL Log

Message explanation

- A resource access was made by the process for which a resource access policy is not defined. The operation was flagged but not denied.
- The operation continued successfully and was not disrupted. A system log has been generated about the missing policy for resource access by the process as denied operation.

Recommended action

- Please contact CISCO TAC with the relevant information as attachments:
  - The message exactly as it appears on the console or in the system log.
  - Output of "show tech-support" (text file).
  - Archive of Btrace files from the box using the following command ("request platform software trace archive target <URL>") For Example: Device#**request platform software trace archive target flash:selinux_btrace_logs**.

SD-WAN on the ESR6300

SD-WAN on the ESR6300 is a network architecture implementation that:

- provides software-defined wide area network capabilities on the ESR6300 platform,
- requires release 17.3.1 or later for the ESR6300 and controller version 20.2 or later, and
- brings the ESR6300 into feature parity with the IR1101.

SD-WAN documentation resources

All of the available SD-WAN documentation can be found here:

https://www.cisco.com/c/en/us/support/routers/sd-wan/series.html

New features for Cisco IOS-XE 17.2.1

This reference provides information about the new features introduced in Cisco IOS-XE 17.2.1 release.

Serial port support

Serial port support is a set of protocol capabilities that:

- enables SCADA Gateway functionality (IEC10x and DNP3),
- provides Raw Socket (TCP and UDP) support,
- includes Line Relay and Reverse Telnet capabilities, and
- brings the ESR6300 into feature compatibility with the IR1101.

Configuration compatibility

All of the configuration and show commands will be the same as are available on the IR1101 platform.

For detailed configuration information, see Cisco Catalyst IR1101 Rugged Series Router Software Configuration Guide.

Boot from the USB

Boot from the USB is a device startup capability that:

- enables booting the device from configuration files located on the pluggable USB,
- supports customized startup configuration files from IOS or from ROMMON, and
- provides flexibility in device initialization and configuration management.

Boot from USB flash drive

This task enables you to boot the router configuration from a USB flash drive, providing flexibility for configuration management and backup scenarios.

The router can boot its configuration from various storage locations including bootflash, flash, NVRAM, USB flash drive, or webui. This procedure shows how to configure USB flash drive as the boot source.

Before you begin

Follow these steps to boot from USB flash drive:

Procedure

Step 1 Display the available boot configuration options.

Example:

```
Router(config)#boot config ?
  bootflash:  URL of the config file
  flash:      URL of the config file
  nvram:      URL of the config file
  usbflash0:  URL of the config file
  webui:      URL of the config file
```

Step 2 Configure the boot command to use the USB flash drive.

Use the syntax: **boot config usbflash0:**<*file name*>

Example:

```
Router(config)#boot config usbflash0:startup-config
Router(config)#
Router#write memory
Building configuration...
[OK]
*Feb 10 10:20:11.990: %SYS-2-PRIVCFG_ENCRYPT: Successfully encrypted private config file
```

Step 3 Verify the boot configuration.

Example:

```
Router#show boot
BOOT variable =
CONFIG_FILE variable = usbflash0:startup-config
BOOTLDR variable does not exist
Configuration register is 0x1820
Standby not ready to show bootvar
```

The CONFIG_FILE variable confirms that the startup-config is set to boot from usbflash0.

The router is now configured to boot from the USB flash drive. The configuration will be loaded from the specified file on the USB flash drive during the next boot cycle.

Boot from ROMMON

This task enables you to configure the router to boot from a USB device when using ROMMON mode.

Follow these steps to boot from the USB:

Procedure

Step 1 Use the **set CONFIG_FILE=usbflash0:***filename* command to specify the configuration file from the ROMMON prompt.

Example:

```
rommon 2 > set CONFIG_FILE=usbflash0:my_startupcfg
           rommon 3 > sync
           rommon 4 > set
           PS1=rommon ! >
           MCU_UPGRADE=SKIP
           THRPUT=
           LICENSE_BOOT_LEVEL=
           RET_2_RTS=
           MCP_STARTUP_TRACEFLAGS=00000000:00000000
           BSI=0
           RANDOM_NUM=1275114933
           BOOT=flash:Jun5_1.SSA,12
           RET_2_RCALTS=951454376
           CONFIG_FILE=usbflash0:my_startupcfg
```

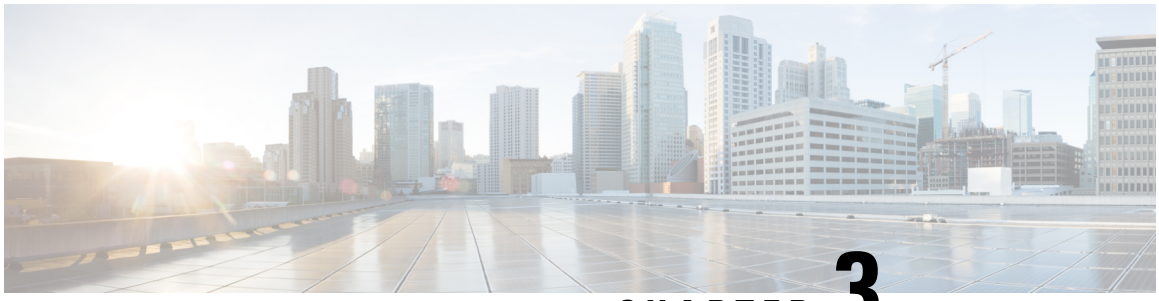
Step 2 Use the ROMMON prompt to continue booting the IOS image as usual.

Boot from USB feature summary

The boot from USB feature provides specific behaviors and characteristics for configuration file management when using USB storage devices.

- Once the CONFIG_FILE is set to a non-default value, the **nvram:startup-config** command is aliased to this new location.
- Any change made to the config file in usbflash will be reflected in nvram:startup-config as well.
- The EXEC command **erase nvram:startup-config** erases the contents of NVRAM, and deletes the file referenced by CONFIG_FILE variable.

- If the USB is unplugged after setting the **boot config usbflash0:**<*filename*> variable, then the day 0 default configuration will take effect.
- When the configuration is saved using the **copy system:running-config nvram:startup-config** command, the device saves a complete version of the configuration file to the location specified by the CONFIG_FILE environment variable, and a distilled version to NVRAM. A distilled version is one that does not contain access list information.



CHAPTER 3

Web User Interface

This section contains the following:

- Web user interface management, on page 29
- Use Web User Interface for Day One setup, on page 29
- Configure LAN settings, on page 30
- Configure primary WAN settings, on page 31
- Configure secondary WAN settings, on page 33
- Configure security settings, on page 34
- GPS mode default enablement, on page 34
- Configure access point name using Cisco WebUI, on page 35
- Review and save router settings, on page 38

Web user interface management

A web user interface is a browser-based management tool that enables you to monitor router performance using an easy-to-read graphical interface.

Use Web User Interface for Day One setup

To configure the web user interface for Day One Setup, you need to perform these tasks:

- Configure the HTTP server.
- Set up the authentication options to log in to the Web UI.
- Access the web user interface.

Follow these steps to configure and access the web user interface for Day One Setup.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip http server** and **ip http secure-server** commands to configure the HTTP server.

Example:

```
Router(config)# ip http server
            Router(config)# ip http secure-server
```

Note
By default, the HTTP server configuration should be present on the device. Verify the configuration by checking that the `ip http server` and `ip http secure-server` commands are present in the running configuration.

Step 3 Set up the authentication options to log in to the Web UI. Use one of these methods to authenticate.

a) Use the **ip http authentication local** command to authenticate using the local database.

Example:

```
Router(config)# ip http authentication local
```

Note
- This command is preconfigured on the device. If the command is not present, configure it as shown in the example.
- You need a user with privilege 15 to access the configuration screens on the Web UI. If the privilege is less than 15, you can access only the Dashboard and Monitoring screens on the Web UI.

b) Use the **username** *username* **privilege** *privilege* **password 0** *passwordtext* command to create a user account.

Example:

```
Router(config)# username <username> privilege <privilege> password 0 <passwordtext>
```

c) Use the **ip http authentication aaa** command to authenticate using AAA options.

Example:

```
Router(config)# ip http authentication aaa
```

Note
Ensure that the required AAA server configuration is present on the device.

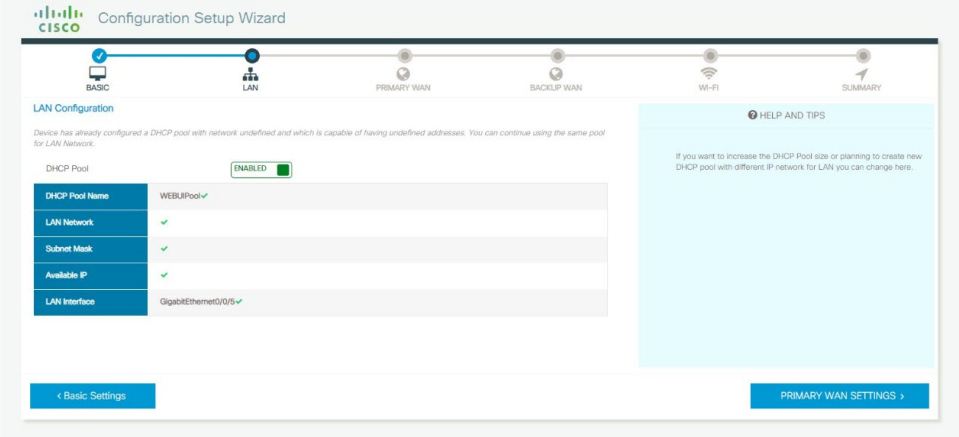
Step 4 Launch a browser and, in the address bar, type the IP address of the device. For a secure connection, type `https://<your-ip-address>/webui/#/dayZeroRouting`.

Step 5 Enter the default username (`cisco`) and the password provided with the device.

Step 6 Click **Log In**.

Configure LAN settings

The following image shows the LAN settings.



Follow these steps to configure the LAN settings.

Procedure

Step 1 Select the interface from the drop-down list.

Step 2 Enter the IP address and subnet mask.

Step 3 Check the **Use this as DHCP Server** check box, and enter the DHCP pool name and network address. Then, select the appropriate time zone from the drop-down list.

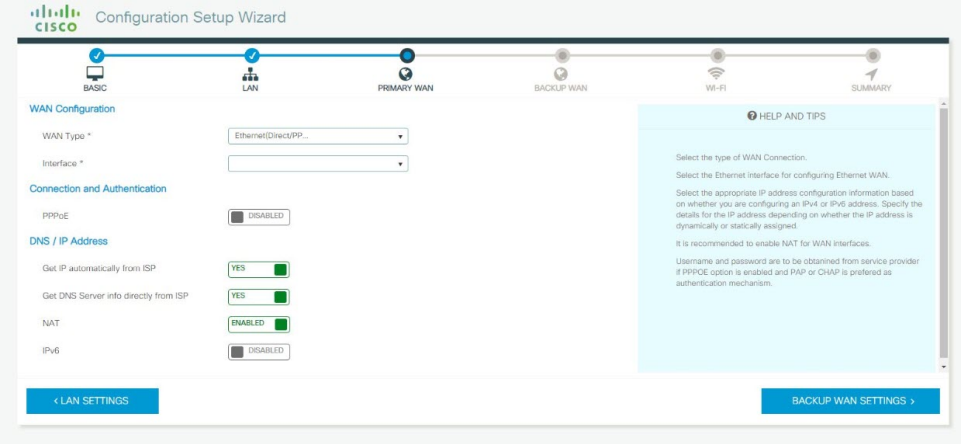
Note
Ensure that you do not use addresses that would conflict with those previously created in the initial configuration dialog.

Step 4 Check the **Configure Management Interface** check box, and enter the IP address of the management interface.

Step 5 Click **Primary WAN Settings**.

Configure primary WAN settings

The following image shows the WAN settings.



Follow these steps to configure the primary WAN settings.

Procedure

Step 1 Select the primary WAN type for the primary WAN connection.

Note
You can configure 3G/4G or Ethernet as a primary WAN, depending on the WAN types supported by the router.

Step 2 Select the interface from the drop-down list.

Step 3 Check the **Get DNS Server info directly from ISP** check box to get the DNS server information directly from the service provider.

Note
You can also manually enter the **Primary DNS** and **Secondary DNS** values.

Step 4 Check the **Get IP automatically from ISP** check box to get the IP address information directly from the service provider.

Note
You can also manually enter the IP address and subnet mask.

Step 5 Check the **Enable NAT** check box to enable NAT.

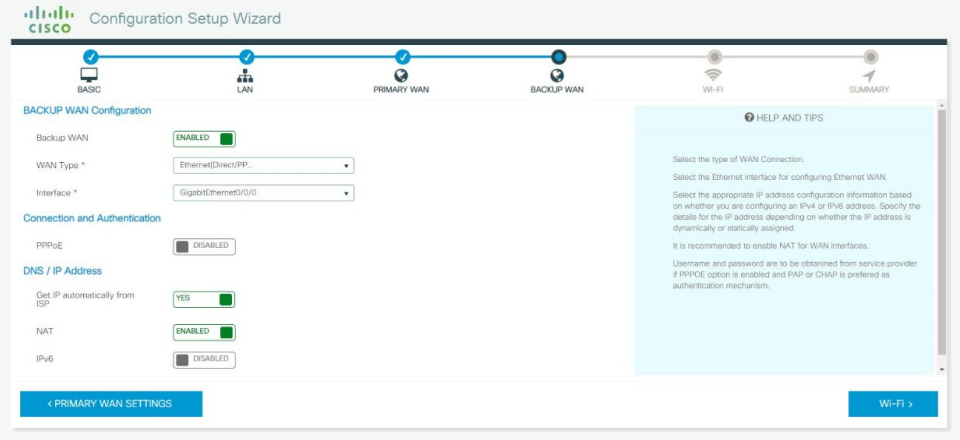
Note
Cisco recommends that you enable NAT.

Step 6 Enter the username and password provided by the service provider.

Step 7 Click **Security / APP Visibility WAN Settings**.

Configure secondary WAN settings

For advanced configuration, you should configure the secondary WAN connection. The following image shows the Backup WAN settings.



Follow these steps to configure the secondary WAN settings.

Procedure

Step 1 Select the secondary WAN type for the secondary WAN connection.

Note
You can configure 3G/4G or Ethernet as a secondary WAN, depending on the WAN types supported by the router.

Step 2 Select the interface from the drop-down list.

Step 3 Check the **Get DNS Server info directly from ISP** check box to get the DNS server information directly from the service provider.

Note
You can also manually enter the **Primary DNS** and **Secondary DNS** values.

Step 4 Check the **Get IP automatically from ISP** check box to get the IP address information directly from the service provider.

Note
You can also manually enter the IP address and subnet mask.

Step 5 Check the **Enable NAT** check box to enable NAT.

Note
Cisco recommends that you enable NAT.

Step 6 Enter the username and password provided by the service provider.

Configure security settings

Note This screen appears only when you select Advanced Mode at first launch.

Follow these steps to configure the security settings.

Procedure

Step 1 Check the **Enable Cisco Recommended Security Settings** check box to ensure that all passwords are encrypted and not shown in plain text.

Step 2 Click **Day 0 Config Summary**.

Step 3 (Optional) To preview the configuration, click **CLI Preview**.

Step 4 Click **Finish** to complete the Day Zero setup.

GPS mode default enablement

GPS mode default enablement is a configuration enhancement that:

- automatically enables GPS mode by default starting with IOS XE 17.9.1,
- sets GPS to standalone mode without requiring additional user configuration, and
- reduces cellular link up time by eliminating the need for manual modem power-cycling.

GPS mode configuration changes

The GPS mode configuration behavior varies between IOS XE versions:

- **Prior to IOS XE 17.9.1**: GPS was enabled by default, but GPS Mode was disabled by default, requiring users to perform an additional modem power-cycle after the router came up to use GPS.
- **Starting with IOS XE 17.9.1**: GPS Mode is enabled by default and set to standalone mode, helping reduce cellular link up time.

Note This only applies to the cellular based GPS. This does not apply to the GPS/GNSS module in IR1800 (DR module), IR8140 (native GPS) and IR8340 (Timing module).

Use the following command to check cellular GPS status:

```
Router# show cellular <slot> gps
auto-reset Enable reset modem automatically after configuring GPS enable or mode
```

Configure access point name using Cisco WebUI

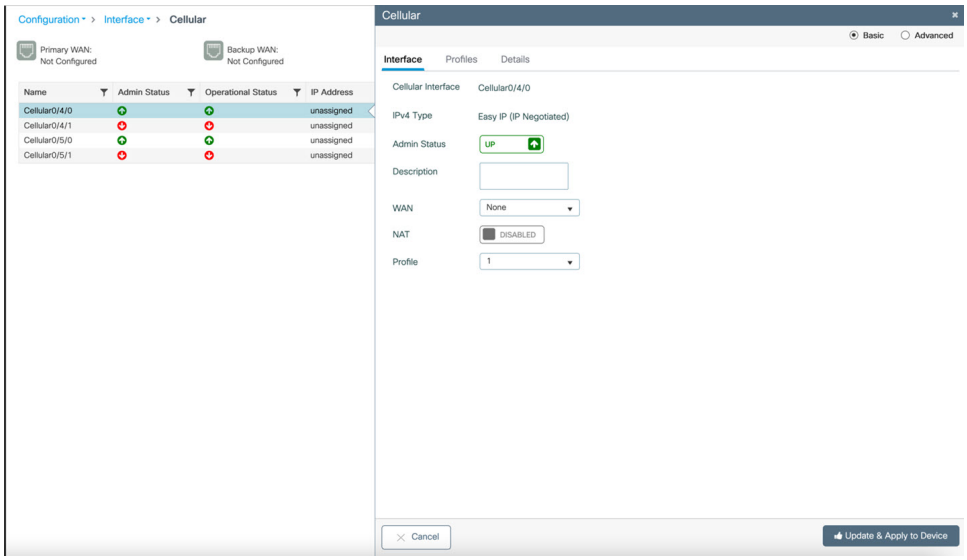
Starting with Cisco IOS XE Release 17.9.1, you can add, edit, or delete the APN from the Cisco WebUI interface. These steps provide an overview of how to perform this function.

Note This section describes only the new functionality and is not a complete overview of the WebUI.

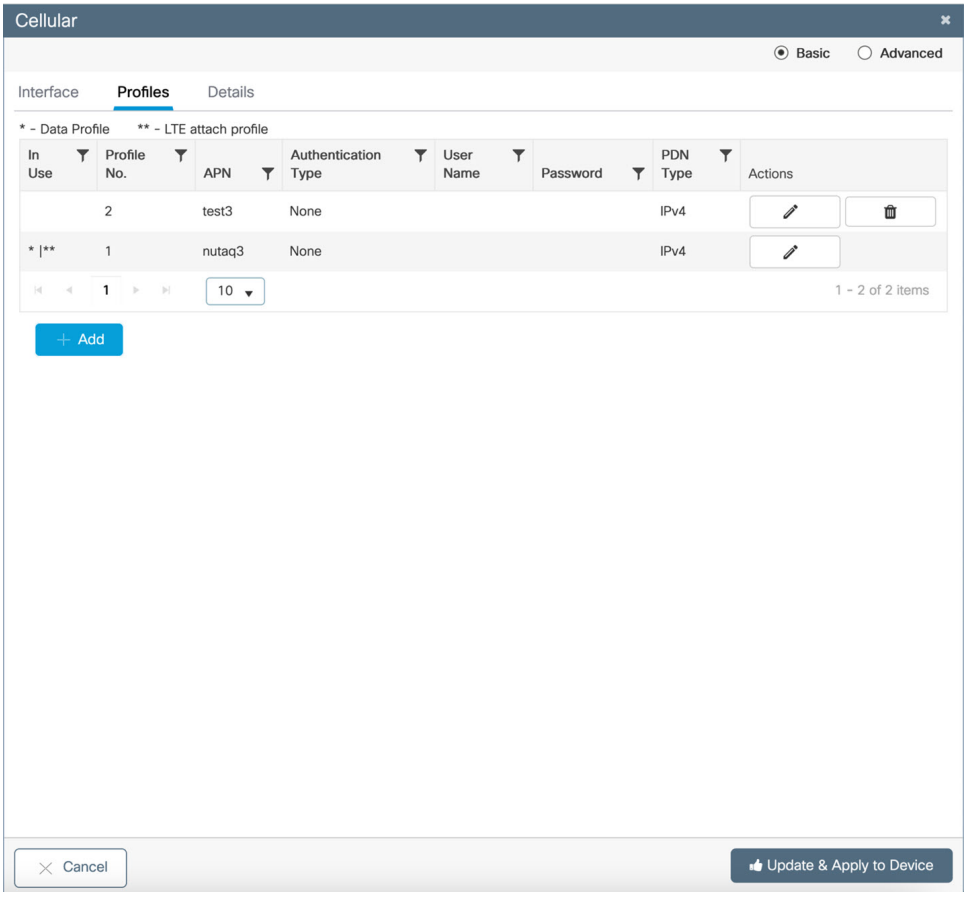
Procedure

Step 1 Navigate to **Configuration > Interface > Cellular**.

Step 2 Double-click the cellular interface based on your platform.



Step 3 On the Cellular window, click the **Profiles** tab.

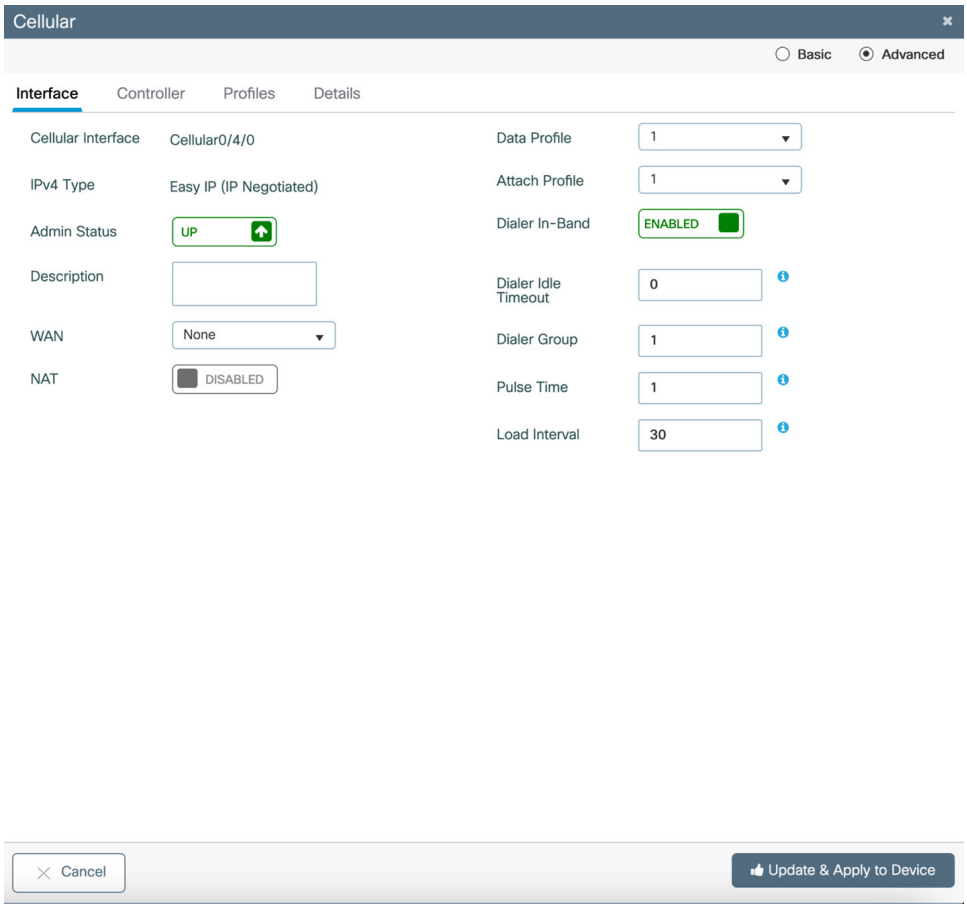


Step 4 From the **Profiles** tab, add, delete, or edit the APN as needed.

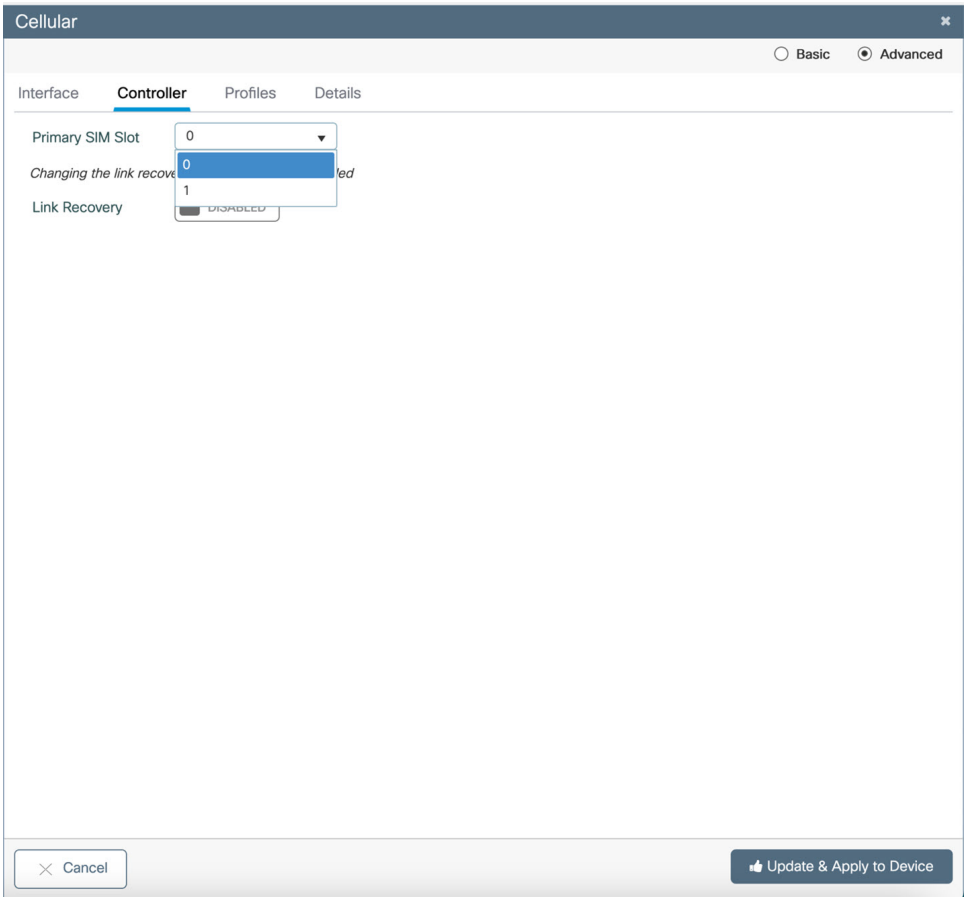
Step 5 Click **Update & Apply to Device** at the bottom of the window.

Step 6 To change the SIM slot, click the **Advanced** radio button at the top of the window.

By default, the APN is attached to SIM slot 0. You can change the APN to SIM slot 1 by using the WebUI.



Step 7 Click the **Controller** tab at the top of the window.



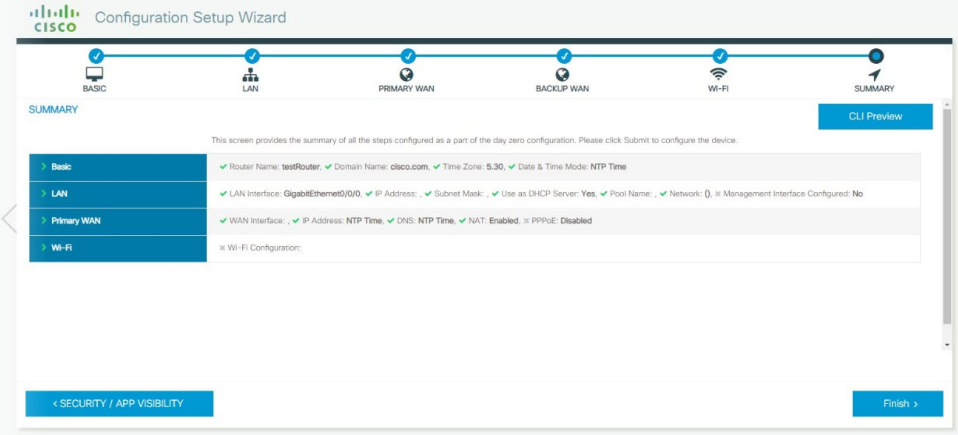
Step 8 From the **Primary SIM Slot** drop-down list, select slot 1.

Step 9 Click **Update & Apply to Device** at the bottom of the window.

Review and save router settings

Refer to the following image for the Router Summary:

Figure 1: Summary Screen

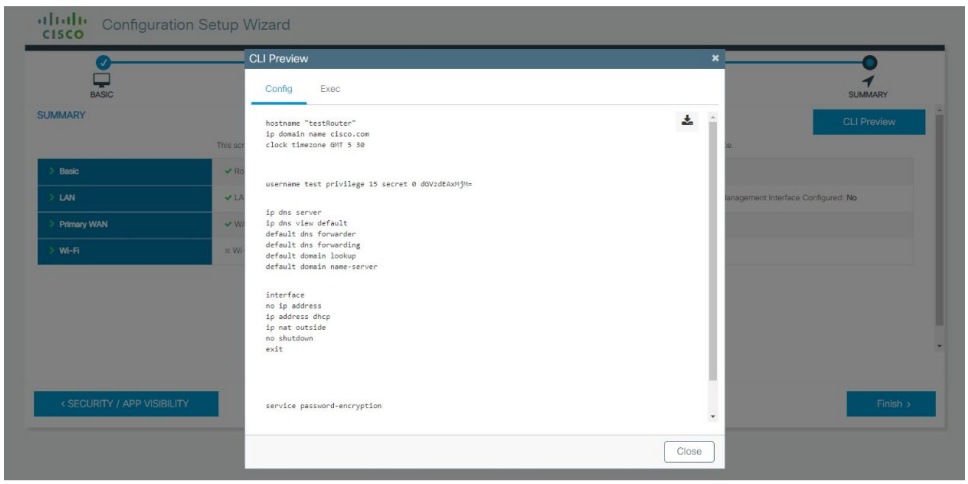


Procedure

Step 1 Review the settings that you have configured so far to verify they are accurate.

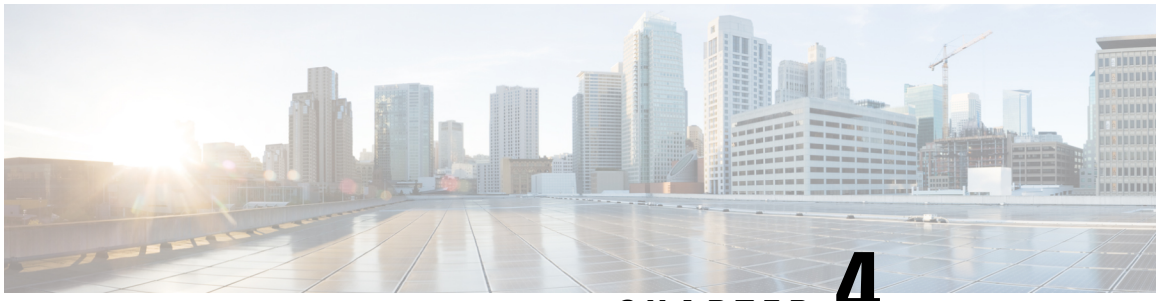
Step 2 If you wish to view the command line entries for your settings, you can click on **CLI Preview**. The following image shows an example:

Figure 2: CLI Settings



Step 3 Close the CLI Preview window.

Step 4 Click **Finish** to save all of your settings.



CHAPTER 4

Secure Shell

This section contains the following topics:

- Secure shell, on page 41
- How to configure secure shell, on page 43
- Secure copy protocol, on page 48
- SSH rekeying, on page 50
- Additional references, on page 55

Secure shell

Secure Shell (SSH) is a protocol that

- provides a secure, remote connection to a device
- provides more security for remote connections than Telnet by providing strong encryption when a device is authenticated, and
- supports SSH Version 1 (SSHv1) and SSH Version 2 (SSHv2) in this software release.

Prerequisites for configure secure shell

The following are the prerequisites for configuring the device for secure shell (SSH):

- For SSH to work, the switch needs an RSA public/private key pair.
- The Secure Shell (SSH) server requires an IPsec (Data Encryption Standard [DES] or 3DES) encryption software image; the SSH client requires an IPsec (DES or 3DES) encryption software image.)
- Configure a hostname and host domain for your device by using the hostname and ip domain-name commands in global configuration mode. Use the **hostname** and **ip domain-name** commands in global configuration mode.

Restrictions for configuring secure shell

Consider these restrictions when configuring the router for secure shell:

- The router supports RSA authentication.

- SSH supports only the execution-shell application.
- The SSH server and the SSH client are supported only on Data Encryption Standard (DES) (56-bit) and 3DES (168-bit) data encryption software. In DES software images, DES is the only encryption algorithm available. In 3DES software images, both DES and 3DES encryption algorithms are available.

Note Cisco highly recommends the 3DES encryption as it is stronger. See the Cisco IOS-XE Device hardening guide at https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html for details.

- This software release supports IP Security (IPSec).
- The router supports the Advanced Encryption Standard (AES) encryption algorithm with a 128-bit key, 192-bit key, or 256-bit key. However, symmetric cipher AES to encrypt the keys is not supported.
- The login banner is not supported in Secure Shell Version 1. It is supported in Secure Shell Version 2, which Cisco recommends due to its better security.
- The -l keyword and userid :{number} {IP-address} delimiter and arguments are mandatory when configuring the alternative method of Reverse SSH for console access.

SSH and router access

Secure Shell (SSH) is a protocol that

- provides a secure, remote connection to a device
- provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated, and
- supports SSH Version 1 (SSHv1) and SSH Version 2 (SSHv2).

SSH protocol characteristics

SSH functions the same in IPv6 as in IPv4. For IPv6, SSH supports IPv6 addresses and enables secure, encrypted connections with remote IPv6 nodes over an IPv6 transport.

SSH servers, integrated clients, and supported versions

The Secure Shell (SSH) Integrated Client feature is an application that

- runs over the SSH protocol to provide device authentication and encryption
- enables a Cisco device to make a secure, encrypted connection to another Cisco device or to any other device running the SSH server, and
- allows for secure communication over an unsecured network with authentication and encryption.

SSH server and client compatibility

The SSH server and SSH integrated client are applications that run on the switch. This connection provides functionality similar to that of an outbound Telnet connection except that the connection is encrypted.

SSH compatibility includes these components:

- SSH server works with the SSH client supported in this release and with non-Cisco SSH clients
- SSH client works with publicly and commercially available SSH servers
- SSH client supports the ciphers of Data Encryption Standard (DES), 3DES, and password authentication



Note The SSH client functionality is available only when the SSH server is enabled.

User authentication is performed like that in the Telnet session to the device. SSH also supports these user authentication methods:

- TACACS+
- RADIUS
- Local authentication and authorization

Recommendation: SSH configuration guidelines

Follow these guidelines when configuring the device as an SSH server or SSH client:

- An RSA key pair generated by a SSHv1 server can be used by an SSHv2 server, and the reverse.
- If you get CLI error messages after entering the **crypto key generate RSA global** configuration command, an RSA key pair has not been generated. Reconfigure the hostname and domain, and then enter the **crypto key generate RSA** command.
- When generating the RSA key pair, the message *No hostname specified* might appear. If it does, you must configure an IP hostname by using the **hostname** global configuration command.
- When generating the RSA key pair, the message No domain specified might appear. If it does, you must configure an IP domain name by using the **IP domain-name** global configuration command.
- When configuring the local authentication and authorization authentication method, make sure that AAA is disabled on the console.

How to configure secure shell

This reference provides the configuration parameters and settings required to establish secure remote connections to network devices using Secure Shell (SSH).

Set up the router to run SSH

SSH provides secure remote access to network devices. This configuration enables encrypted communication and helps protect against unauthorized access attempts.



Note Ensure that you configure user authentication for local or remote access.

Follow these steps to set up the router to run SSH.

Procedure

Step 1 Use the **enable** command to enter privileged EXEC mode.

Example:

```
Router> enable
```

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 3 Use the **hostname** *your_hostname* and **ip domain-name** *your_domain_name* commands to configure a hostname and domain name for the device.

Example:

```
Router(config)# hostname <your_hostname>
Router(config)# ip domain-name <your_domain_name>
```

Step 4 Use the **crypto key generate rsa** command to generate an RSA key pair to enable SSH.

Example:

```
Router(config)# crypto key generate rsa
```

Note
Use a modulus size of at least 1024 bits for security.

Step 5 Use the **end** command to exit global configuration mode.

Example:

```
Router(config)# end
```

Step 6 Use the **show running-config** command to verify the configuration.

Example:

```
Router# show running-config
```

Step 7 (Optional) Use the **copy running-config startup-config** command to save the configuration.

Example:

```
Router# copy running-config startup-config
```

The router is now configured to accept SSH connections. You can connect remotely using an SSH client with the configured hostname and the appropriate user credentials.

Configure the SSH server

Configure the SSH server on a device to enable secure remote CLI access and to control SSH protocol parameters. Use this procedure when you need to set up the device as an SSH server to provide secure command-line interface access for remote management.

Note This procedure is required only if you are configuring the device as an SSH server.

Follow these steps to configure the SSH server.

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 3 (Optional) Use the **ip ssh version** [**2**] command to configure the device to run SSH Version 2.

Example:

```
Router(config)# ip ssh version 2
```

Note
If you do not enter this command or do not specify a keyword, the SSH server selects the latest SSH version supported by the SSH client. For example, if the SSH client supports SSHv1 and SSHv2, the SSH server selects SSHv2.

Step 4 Use the **ip ssh** {**timeout** *seconds* | **authentication-retries** *number*} command to configure the SSH control parameters.

Example:

```
Router(config)# ip ssh timeout 90
Router(config)# ip ssh authentication-retries 2
```

Note

- Specify the timeout value in seconds; the default is 120 seconds, and the range is 0 to 120 seconds. This parameter applies to the SSH negotiation phase. After the connection is established, the device uses the default timeout values of the CLI-based sessions.
- By default, up to five simultaneous, encrypted SSH connections for multiple CLI-based sessions over the network are available (session 0 to session 4). After the execution shell starts, the CLI-based session timeout value returns to the default of 10 minutes.

- Specify the number of times that a client can re-authenticate to the server. The default is 3, and the range is 0 to 5.
- Repeat this step when configuring both parameters.

Step 5 (Optional) Use the **line vty** *line-number* [*ending-line-number*] command to enter line configuration mode and configure the virtual terminal line settings. Then, use the **transport input ssh** command to limit the device to SSH connections only.

Example:

```
Router(config)# line vty 1 10
Router(config-line)# transport input ssh
```

Note

- For the *line-number* and *ending-line-number* arguments, the range is from 0 to 15.
- The `transport input ssh` command specifies that the device prevents non-SSH Telnet connections, limiting the device to only SSH connections.

Step 6 Use the **end** command to exit line configuration mode and return to privileged EXEC mode.

Example:

```
Router(config-line)# end
```

Step 7 Use the **show running-config** command to verify your entries.

Example:

```
Router# show running-config
```

Step 8 (Optional) Use the **copy running-config startup-config** command to save your entries in the configuration file.

Example:

```
Router# copy running-config startup-config
```

The SSH server is configured and ready to accept secure connections with the specified parameters for version, timeout, and authentication settings.

Monitor the SSH configuration and status

This reference provides commands to monitor and display SSH server configuration and status information.

Table 1: Commands for displaying the SSH server configuration and status

Command	Purpose
show ip SSH	Shows the version and configuration information for the SSH server.
show SSH	Shows the status of the SSH server.

Configure the router for local authentication and authorization

You can configure AAA to operate without a server by setting the router to implement AAA in local mode. The router then handles authentication and authorization. No accounting is available in this configuration.



Note To secure the router for HTTP access by using AAA methods, you must configure the router with the `ip http authentication aaa` global configuration command. Configuring AAA authentication alone does not secure the router for HTTP access.

Follow these steps to configure AAA to operate without a server by setting the router to implement AAA in local mode.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **aaa new-model** command to enable AAA.

Example:

```
Router(config)# aaa new-model
```

Step 3 Use the **aaa authentication login default local** command to set the login authentication to use the local username database.

Example:

```
Router(config)# aaa authentication login default local
```

Note
The `default` keyword applies the local user database authentication to all ports.

Step 4 Use the **aaa authorization exec local** command to configure user AAA authorization, check the local database, and allow the user to run an EXEC shell.

Example:

```
Router(config)# aaa authorization exec local
```

Step 5 Use the **aaa authorization network local** command to configure user AAA authorization for all network-related service requests.

Example:

```
Router(config)# aaa authorization network local
```

Step 6 Use the **username** *name* [**privilege** *level*] **password** *encryption-type password* command to enter the local database and establish a username-based authentication system.

Example:

```
Router(config)# username your_user_name privilege 1 password 7 secret567
```

Note

- Repeat this command for each user.
- For *name*, specify the user ID as one word. Spaces and quotation marks are not allowed.
- (Optional) For *level*, specify the privilege level the user has after gaining access. The range is 0 to 15. Level 15 gives privileged EXEC mode access; level 0 gives user EXEC mode access.
- For *encryption-type*, enter 0 to specify that an unencrypted password follows, or enter 7 to specify that a hidden password follows.
- For *password*, specify the password the user must enter to gain access to the router. The password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the `username` command.

Step 7 Use the **end** command to exit configuration mode and return to privileged EXEC mode.

Example:

```
Router(config)# end
```

Step 8 Use the **show running-config** command to verify your entries.

Example:

```
Router# show running-config
```

Step 9 (Optional) Use the **copy running-config startup-config** command to save your entries in the configuration file.

Example:

```
Router# copy running-config startup-config
```

The router is configured for local authentication and authorization using AAA methods. Users can authenticate using the local username database, and authorization is handled locally by the router.

Secure copy protocol

The Secure Copy Protocol (SCP) is a network protocol that

- provides a secure and authenticated method for copying router configuration or router image files
- relies on Secure Shell (SSH), an application and a protocol that provide a secure replacement for the Berkeley r-tools, and
- enables secure file transfer operations between network devices.

Requirements for secure copy configuration

Before enabling SCP, you must correctly configure SSH, authentication, and authorization on the switch.

- Because SCP relies on SSH for its secure transport, the router must have an RSA key pair.
- SCP relies on SSH for security.

- SCP requires that authentication, authorization, and accounting (AAA) authorization be configured so the router can determine whether the user has the correct privilege level.
- A user must have appropriate authorization to use SCP.
- A user who has appropriate authorization can use SCP to copy any file in the Cisco IOS File System (IFS) to and from a switch by using the **copy** command. An authorized administrator can also do this from a workstation.

Restrictions for configuring secure copy

Before enabling SCP, you must correctly configure SSH, authentication, and authorization on the router.

When using SCP, you cannot enter the password into the **copy** command. You must enter the password when prompted.

Configure secure copy

This task enables SCP server functionality on a Cisco router to allow secure file transfers between the router and remote hosts. SCP provides a secure method for copying files to and from network devices.



Note Before you configure SCP, ensure that proper authentication mechanisms are in place.

Follow these steps to configure the Cisco router for Secure Copy (SCP) server-side functionality.

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 3 Use the **aaa new-model** command to enable the AAA access control model.

Example:

```
Router(config)# aaa new-model
```

Step 4 Use the **aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] command to set AAA authentication at login.

Example:

```
Router(config)# aaa authentication login default group tacacs+
```

Step 5 Use the **username** *name* [**privilege** *level*] **password** *encryption-type encrypted-password* command to establish a username-based authentication system.

Example:

```
Router(config)# username superuser privilege 2 password 0 superpassword
```

Note
You can omit this step if a network-based authentication mechanism, such as TACACS+ or RADIUS, is configured.

Step 6 Use the **ip scp server enable** command to enable SCP server-side functionality.

Example:

```
Router(config)# ip scp server enable
```

Step 7 Use the **exit** command to exit global configuration mode and return to privileged EXEC mode.

Example:

```
Router(config)# exit
```

Step 8 (Optional) Use the **show running-config** command to display the SCP server-side functionality.

Example:

```
Router# show running-config
```

Step 9 (Optional) Use the **debug ip scp** command to troubleshoot SCP authentication problems.

Example:

```
Router# debug ip scp
```

SCP server functionality is now enabled on the router, allowing secure file transfers. You can now use SCP commands to copy files to and from the router.

```
Router# copy scp <somefile> your_username@remotehost:/<some/remote/directory>
```

SSH rekeying

SSH rekeying is a security feature that:

- replaces the encryption and integrity keys during an active SSH session
- initiates a new key exchange based on a configured time or data threshold, and
- helps reduce the risk of long-term key exposure.

SSH rekeying characteristics

SSH rekeying can be initiated by either the SSH client or server. During rekeying, data transfer halts until the key exchange completes. The encryption algorithm negotiated at session startup remains in use during rekeying. OpenSSH-based clients version 7.5 or later support SSH rekey exchange.

SSH rekeying scenarios

- Rekeying every hour to ensure session security over long connections.
- Rekeying after 1 GB of data transfer to limit the data encrypted with a single key.

SSH rekeying types

SSH rekeying types are configuration methods that:

- enable periodic renewal of encryption keys during SSH sessions
- provide time-based and volume-based triggering options using the **ip SSH rekey** command, and
- enhance security by preventing prolonged use of the same encryption keys.

Configuration methods

You can configure SSH rekeying on Cisco routers in two ways by using the **ip SSH rekey** command:

- Time-based rekeying: Specify the interval (in minutes or seconds) after which rekeying occurs.
- Volume-based rekeying: Specify the data volume (in KB or MB) that triggers rekeying.

By default the **ip SSH rekey** command is disabled. If you configure these settings using the **ip SSH rekey** command without specifying time or volume, then it will apply both time and volume rekey with default values. The default value for time is 60 minutes and volume is 1 GB.

Rekeying by time initiates a key exchange after a specific time interval:

- In Common Criteria mode, the maximum allowed time is less than 60 minutes or 3600 seconds.
- The configurable range is 10 to 1440 minutes.
- For some devices, the default is 60 minutes, and the command accepts values in minutes.

Rekeying by volume enforces a new key exchange when the transmitted or received data reaches a configured threshold:

- In Common Criteria mode, the maximum rekey volume is 1024 MB (1 GB).
- The configurable range for the rekey volume is 100 KB to 4 GB.
- For some devices, the default is 1 GB, and the command accepts values in kilobytes.

Prerequisites for configuring SSH rekeying

To configure SSH rekeying on Cisco routers, ensure these prerequisites are met.

- The device runs Cisco IOS with SSH enabled.
- SSH version 2 is recommended.
- Administrative privilege access is required.

Best practices for SSH rekeying configuration

Best practices for SSH rekeying configuration are:

- Enable SSH rekeying to reduce the risk from potential key compromise, especially in high-security environments.
- Ensure rekey intervals comply with security policies, such as Common Criteria, by setting appropriate time or volume thresholds.

SSH session idle timeout settings

SSH sessions can be configured to disconnect after a period of inactivity.

- SSH idle timeout: **line vty <0-240>** (in minutes; 0 means never timeout)
- **exec-timeout <0-240>** (in minutes; 0 means never timeout)

Restrictions for SSH rekeying on Cisco routers

The following limitations apply to SSH rekeying on Cisco routers:

- During SCP downloads, the device acting as receiver cannot control the sender's data volume, so rekey is triggered only after the configured volume is received.
- Large SSH bulk mode settings (for example, 1 GB) can lead to fewer rekeys than expected due to larger data windows.
- SSH sessions established without rekey configuration do not support rekeying
- When rekey is enabled, existing sessions will only adopt rekeying after the next exchange initiated from the other side.
- Removing the rekey configuration disables SSH rekey for existing sessions.
- Changing the rekey configuration does not affect active sessions until the subsequent rekey exchange.
- When a rekey exchange occurs, the related data and time parameters for the session reset to the new configured value.
- Do not modify the encryption method during rekeying.

The actual number of rekeys during SCP downloads may not match calculated expectations.

This behavior is by design and aligns with current SSH implementation.

Example observation:

Transferring a 631.3 MB file with a 100 KB rekey threshold could yield an expected 6,463 rekeys, but only 689 may occur.

This discrepancy is documented and not considered a defect.

Configure SSH rekeying

SSH rekeying enhances session security by periodically renewing encryption keys. Rekeying can be based on time intervals or data volumes. The configuration involves enabling SSH, setting the SSH version, and defining the rekey parameters.

Follow these steps to configure SSH rekeying.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip domain-name** *domain-name* command to set the domain name.

Example:

```
Router(config)# ip domain-name example.com
```

Step 3 Use the **crypto key generate rsa modulus** *modulus-size* command to generate the RSA keys.

Example:

```
Router(config)# crypto key generate rsa modulus 2048
```

Step 4 Use the **ip ssh version 2** command to set the SSH version to 2.

Example:

```
Router(config)# ip ssh version 2
```

Step 5 Use the **ip ssh rekey time** *minutes* command to configure time-based rekey.

Example:

```
Router(config)# ip ssh rekey time 120
```

Note
The valid range is 10 to 1440 minutes (the default is 60 minutes).

Step 6 Use the **ip ssh rekey volume** *size-in-kb* command to configure volume-based rekey.

Example:

```
Router(config)# ip ssh rekey volume 10240
```

Note

- This example triggers a rekey after 10,240 KB.
- The valid range is 100 to 4194303 KB (the default is 1 GB).

Step 7 (Optional) Use the **no ip ssh rekey time** or **no ip ssh rekey volume** command to disable time-based or volume-based rekeying.

Example:

```
Router(config)# no ip ssh rekey time
```

Example:

```
Router(config)# no ip ssh rekey volume
```

SSH rekeying is enabled on the device, improving session security through periodic key renewal.

Verify SSH rekey configuration

Verification ensures that SSH rekeying is correctly configured and operational on the device.

Follow these steps to verify the SSH rekey configuration.

Procedure

Step 1 Use the **show ip ssh** command to display the SSH configuration details.

Example:

```
Router# show ip ssh
            SSH Enabled - version 2.0
            Authentication methods:publickey,keyboard-interactive,password
            Authentication Publickey
Algorithms:ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-ed25519,x509v3-ecdsa-sha2-nistp256,x509v3-ecdsa-sha2-nistp384,x509v3-ecdsa-sha2-nistp521,rsa-sha2-256,rsa-sha2-512,x509v3-rsa2048-sha256

            Hostkey
Algorithms:ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,rsa-sha2-512,rsa-sha2-256

            Encryption
Algorithms:chacha20-poly1305@openssh.com,aes128-gcm@openssh.com,aes256-gcm@openssh.com,aes128-gcm,aes256-gcm,aes128-ctr,aes192-ctr,aes256-ctr

            MAC
Algorithms:hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-512

            KEX
Algorithms:curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha256,diffie-hellman-group16-sha512

            Authentication timeout: 120 secs; Authentication retries: 3
            Minimum expected Diffie Hellman key size : 2048 bits
            IOS Keys in SECSH format(ssh-rsa, base64 encoded): TP-self-signed-1538213291
            Modulus Size : 2048 bits
           ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCw/l05OHVk/3MAC30fIpLf6rphTETPZtnIdlYbZPCc

           L+gLy+dxsxq8kYXZ5lTQq+lstB+PB868xDz/mkk9OfNFcEwa2SavoyHuDeqN58FMFmU4z9guwfe66dIk

           sdt1qA086Wc45BUBYrE3R4uVC1gQdkuHyIZmN54c3lvSRuToaGNo3M10rRjIELOWOV1Qbcr8e0nrAAmf

           Xznx2lDqO2QPdynLEpLWinLpnwEFEJtQ2DXRAZO+a1SqGvfTbLZ+EQ7SCI39X0xC5YVCFRjD3NF+6NpO

            Q6wt0my/5xNXr1m5uabfOco96lAXbCD4w55JDuv0Eq3siefVAuXXtgGpDCVB
            IOS Keys in SECSH format(ssh-ec, base64 encoded): NONE
```

Note

The SSH session attributes and algorithms include:

- SSH version and enablement status
- Supported authentication methods and algorithms (public key, keyboard-interactive, password)
- Hostkey and encryption algorithms

- MAC and key exchange algorithms
- Authentication timeout and retry settings
- Minimum expected Diffie-Hellman key size
- Current session keys (in SECSH format)

Step 2 Use the **show running-config | include ssh** command to display the SSH-related running configuration details.

Example:

```
Router# show running-config | include ssh
            ip ssh bulk-mode 13107245
            ip ssh rekey volume 100
            transport input ssh
            transport input ssh
```

Example:

```
Router# show running-config | include ssh
            ip ssh bulk-mode 13107245
            ip ssh rekey time 120
            transport input ssh
            transport input ssh
```

Step 3 Use the **show ssh rekey** command to monitor SSH rekey activity and current session information.

Example:

```
Router# show ssh rekey
            SSHv2 server sessions
            Connection  TimeTillRekey  DataLeftTillRekey TimeRekeys  VolumeRekeys
            0           0:00           98                0           0

            %No SSHv2 client connections running.
```

Example:

```
Router# show ssh rekey
            SSHv2 server sessions
            Connection  TimeTillRekey  DataLeftTillRekey TimeRekeys  VolumeRekeys
            0           0:00           99997             0           0

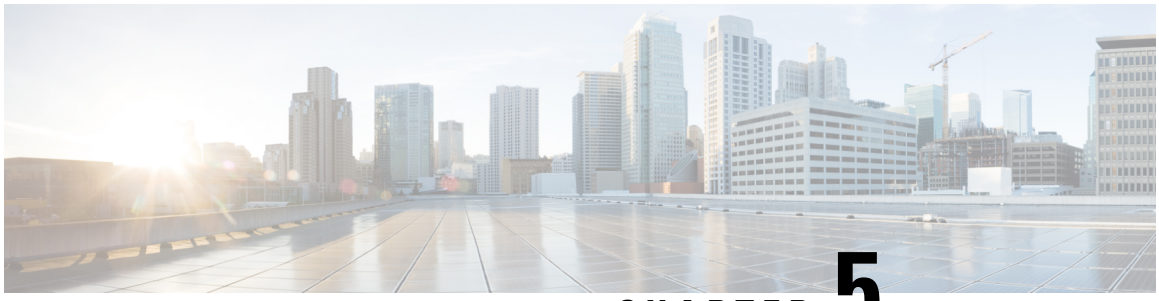
            SSHv2 client sessions
            Connection  TimeTillRekey  DataLeftTillRekey TimeRekeys  VolumeRekeys
            1           0:00           0                 0           0
```

Additional references

This sections provide references related to the SSH feature.

Related Topic	Document Title
Configuring Identity Control policies and Identity Service templates for Session Aware networking.	Session Aware Networking Configuration Guide, Cisco IOS XE Release 3SE: https://www.cisco.com/en/US/docs/ios-xml/ios/san/configuration/xe-3se/3850/san-xe-3se-3850-book.pdf

Related Topic	Document Title
Configuring RADIUS, TACACS+, Secure Shell, 802.1X and AAA.	Secure Shell Configuration Guide, Cisco IOS XE Gibraltar 16.11.x: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9500/software/release/16-11/configuration_guide/sec/b_1611_sec_9500_cg/configuring_secure_shell__ssh_.html



CHAPTER 5

Using Cisco IOS XE Software

This chapter contains the following sections:

- Command modes, on page 57
- Use keyboard shortcuts, on page 59
- No and default command forms, on page 59
- Use the history buffer to recall commands, on page 60
- Configuration files management, on page 60
- Save configuration changes, on page 61
- Filtering output from show and more commands, on page 61
- Cisco feature navigator, on page 62
- Find support information for platforms and Cisco software images, on page 62
- Get help, on page 63
- Find command options, on page 63
- Software advisor, on page 67
- Use software release notes, on page 67

Command modes

The command modes available in Cisco IOS XE are the same as those available in traditional Cisco IOS. Use the CLI to access Cisco IOS XE software. Because the CLI is divided into many different modes, the commands available to you at any given time depend on the mode that you are currently in. Entering a question mark (**?**) at the CLI prompt allows you to obtain a list of commands available for each command mode.

When you log in to the CLI, you are in user EXEC mode. User EXEC mode contains only a limited subset of commands. To have access to all commands, you must enter privileged EXEC mode, normally by using a password. From privileged EXEC mode, you can issue any EXEC command—user or privileged mode—or you can enter global configuration mode. Most EXEC commands are one-time commands. For example, **show** commands show important status information, and **clear** commands clear counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and a variety of other modes, such as protocol-specific modes.

ROM monitor mode is a separate mode used when the Cisco IOS XE software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode.

Table 2: Accessing and exiting command modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** command.	`Router#`	To return to user EXEC mode, use the **disable** command.
Global configuration	From privileged EXEC mode, use the **configure termina**l command.	`Router(config)#`	To return to privileged EXEC mode from global configuration mode, use the **exit** or **end** command.
Interface configuration	From global configuration mode, specify an interface using an **interface** command.	`Router(config-if)#`	To return to global configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command.
Diagnostic	The router boots up or accesses diagnostic mode in these scenarios: • In some cases, diagnostic mode will be reached when the Cisco IOS process or processes fail. In most scenarios, however, the router will reload. • A user-configured access policy is configured using the **transport-map** command that directs a user into diagnostic mode. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) is entered and the router is configured to go to diagnostic mode when the break signal is received.	`Router(diag)#`	If failure of the Cisco IOS process is the reason for entering diagnostic mode, the Cisco IOS problem must be resolved and the router rebooted to get out of diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or by using a method that is configured to connect to the Cisco IOS CLI.
ROM monitor	From privileged EXEC mode, use the **reload** EXEC command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon#>`	To exit ROM monitor mode, manually boot a valid image or perform a reset with autoboot set so that a valid image is loaded.

Use keyboard shortcuts

Commands are not case sensitive. You can abbreviate commands and parameters if the abbreviations contain enough letters to be different from any other currently available commands or parameters.

Table 3: Keyboard shortcuts

Key Name	Purpose
Ctrl-B or the **Left Arrow** key	Move the cursor back one character.
Ctrl-F or the **Right Arrow** key	Move the cursor forward one character.
Ctrl-A	Move the cursor to the beginning of the command line.
Ctrl-E	Move the cursor to the end of the command line.
Esc B	Move the cursor back one word.
Esc F	Move the cursor forward one word.

No and default command forms

No and default command forms are CLI command variations that

- provide options to disable, enable, or reset configuration functions
- apply to almost every configuration command in Cisco IOS software, and
- allow administrators to modify command behavior without entering complete command syntax.

Command form functions

The **no** form is used to disable a function. Use the command without the **no** keyword to re-enable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no IP routing** command; to re-enable IP routing, use the **IP routing** command. The Cisco IOS software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

By issuing the **<command> default** command-name, you can configure the command to its default setting. The Cisco IOS software command reference publications describe the function from a **default** form of the command when the **default** form performs a different function than the plain and **no** forms of the command. To see what default commands are available on your system, enter **default ?** in the appropriate command mode.

Use the history buffer to recall commands

The history buffer stores the last 20 commands you entered. History substitution allows you to access these commands without retyping them, by using special abbreviated commands.

Table 4: History substitution commands

Command	Purpose
Ctrl-P or the **Up Arrow** key	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Ctrl-N or the **Down Arrow** key[1]	Returns to more recent commands in the history buffer after recalling commands with **Ctrl-P** or the **Up Arrow** key.
Router# show history	While in EXEC mode, lists the last few commands you entered.

Configuration files management

Managing configuration files is a router maintenance process that:

- stores startup configuration files in the nvram: file system and running configuration files in the system: file system
- provides encryption of configuration files through IOS XE features, and
- enables backup of configuration files to prevent data loss.

Configuration file storage and management

This configuration file storage setup is also used on several other Cisco router platforms.

IOS XE provides encryption of the configuration file. Encryption is discussed in length in the IOS XE hardening device guide which can be found here:https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html

As a matter of routine maintenance on any Cisco router, users should back up the startup configuration file by copying the startup configuration file from NVRAM to one of the router's other file systems and, additionally, to a network server. Backing up the startup configuration file provides an easy method of recovering the startup configuration file if the startup configuration file in NVRAM becomes unusable for any reason.

The **copy** command can be used to back up startup configuration files.

For more detailed information on managing configuration files, see the "Managing Configuration Files" section in the Cisco IOS XE Configuration Fundamentals Configuration Guide.

Save configuration changes

Save configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs.

This task saves the configuration to the NVRAM.

Procedure

Use the **copy running-config startup-config** command to save your configuration changes to the startup configuration.

Example:

```
Router# copy running-config startup-config
Destination filename [startup-config]? enter
Building configuration...
[OK]
Router#
*Sep 24 08:50:26.666: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config
 file
```

Note
It may take a few minutes to save the configuration.

The configuration is saved to the NVRAM and changes persist through software reloads and power outages.

Filtering output from show and more commands

Filtering output from show and more commands is a functionality that

- searches and filters the output of **show** and **more** commands
- helps you sort through large amounts of output, and
- excludes output that you need not see.

Command syntax

To use this functionality, enter a **show** or **more** command followed by the "pipe" character (|); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case sensitive):

show *command* | **{append | begin | exclude | include | redirect | section | tee}** *regular-expression*

The output matches certain lines of information in the configuration file.

The example for show interface command with include protocol modifier is as given.

In this example, a modifier of the **show interface** command (**include protocol**) is used to provide only the output lines in which the expression **protocol** is displayed:

```
Router# show interface | include protocol
GigabitEthernet0/0/0 is administratively down, line protocol is down
     0 unknown protocol drops
GigabitEthernet0/0/1 is administratively down, line protocol is down
     0 unknown protocol drops
GigabitEthernet0/1/0 is administratively down, line protocol is down (disabled)
     0 unknown protocol drops
GigabitEthernet0/1/1 is administratively down, line protocol is down (disabled)
     0 unknown protocol drops
GigabitEthernet0/1/2 is administratively down, line protocol is down (disabled)
     0 unknown protocol drops
GigabitEthernet0/1/3 is administratively down, line protocol is down (disabled)
     0 unknown protocol drops
Cellular0/3/0 is administratively down, line protocol is down
     0 unknown protocol drops
Cellular0/3/1 is down, line protocol is down
     0 unknown protocol drops
Async0/2/0 is administratively down, line protocol is down
     0 unknown protocol drops
Vlan1 is up, line protocol is down , Autostate Enabled
     0 unknown protocol drops
Vlan5 is administratively down, line protocol is down , Autostate Enabled
     0 unknown protocol drops
Vlan57 is up, line protocol is down , Autostate Enabled
     0 unknown protocol drops
```

Cisco feature navigator

Cisco Feature Navigator is a tool that:

- enables you to determine which Cisco IOS XE software images support a specific software release, feature set, or platform
- provides information about platform support and software image support, and
- does not require an account on Cisco.com to use.

Find support information for platforms and Cisco software images

The Cisco IOS XE software is packaged in feature sets consisting of software images that support specific platforms. The group of feature sets that are available for a specific platform depends on which Cisco software images are included in a release.

All of the Cisco IOS-XE configuration guides can be found here: https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xe-17/series.html.

To identify the set of software images available in a specific release or to find out if a feature is available in a given Cisco IOS XE software image, you can use Cisco Feature Navigator or see the https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xe-17/series.html.

Get help

Entering a question mark (**?**) at the CLI prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To get help that is specific to a command mode, a command, a keyword, or an argument, use one of these commands.

Command	Purpose
help	Provides a brief description of the help system in any command mode.
abbreviated-command-entry ?	Provides a list of commands that begin with a particular character string. **Note** There is no space between the command and the question mark.
abbreviated-command-entry <Tab>	Completes a partial command name.
?	Lists all the commands that are available for a particular command mode.
command ?	Lists the keywords or arguments that you must enter next on the command line. **Note** There is a space between the command and the question mark.

Find command options

This section provides information about how to display the syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (**?**) at the configuration prompt or after entering a part of a command followed by a space. The Cisco IOS XE software displays a list and brief descriptions of the available keywords and arguments. For example, if you are in global configuration mode and want to see all the keywords and arguments for the **arap** command, you should type **arap ?**.

The <**cr**> symbol in command help output stands for carriage return. On older keyboards, the carriage return key is the **Return** key. On most modern keyboards, the carriage return key is the **Enter** key. The <**cr**> symbol at the end of command help output indicates that you have the option to press **Enter** to complete the command and that the arguments and keywords in the list preceding the <**cr**> symbol are optional. The <**cr**> symbol by itself indicates that no more arguments or keywords are available, and that you must press **Enter** to complete the command.

Table 5: Finding command options

Command	Comment

`Router> enable`
`Password: <password>`
`Router#`
	Enter the **enable** command and password to access privileged EXEC commands. You are in privileged EXEC mode when the prompt changes to a " # " from the " > ", for example, **Router>** to **Router#**

`Router# configure terminal`
`Enter configuration commands, one per line. End with CNTL/Z.`
`Router(config)#`
	Enter the **configure terminal** privileged EXEC command to enter global configuration mode. You are in global configuration mode when the prompt changes to **Router (config)#**

`Router(config)# interface GigabitEthernet ?`
`<0-0> GigabitEthernet interface number`
 	
`Router(config)# interface GigabitEthernet 0/?`
`<0-5> Port Adapter number`
 	
`Router (config)# interface GigabitEthernet 0/0/?`
`<0-63> GigabitEthernet interface number`
 	
`Router (config)# interface GigabitEthernet 0/0/0?`
`.  <0-71>`
 	
`Router(config-if)#`
	Enter interface configuration mode by specifying the interface that you want to configure, using the **interface GigabitEthernet** global configuration command.
	
	Enter **?** to display what you must enter next on the command line.
	
	When the <cr> symbol is displayed, you can press **Enter** to complete the command.
	
	You are in interface configuration mode when the prompt changes to **Router(config-if)#**

Command	Comment
Router(config-if)# **?** Interface configuration commands: . . . ip Interface Internet Protocol config commands keepalive Enable keepalive lan-name LAN Name command llc2 LLC2 Interface Subcommands load-interval Specify interval for load calculation for an interface locaddr-priority Assign a priority group logging Configure logging for interface loopback Configure internal loopback on an interface mac-address Manually set interface MAC address mls mls router sub/interface commands mpoa MPOA interface configuration commands mtu Set the interface Maximum Transmission Unit (MTU) netbios Use a defined NETBIOS access list or enable name-caching no Negate a command or set its defaults nrzi-encoding Enable use of NRZI encoding ntp Configure NTP . . . Router(config-if)#	Enter **?** to display a list of all the interface configuration commands available for the interface. This example shows only some of the available interface configuration commands.

Command	Comment

`Router(config-if)# ip ?`
`Interface IP configuration subcommands:`
`  access-group  Specify access control for packets`
`  accounting  Enable IP accounting on this interface`
`  address  Set the IP address of an interface`
`  authentication  authentication subcommands`
`  bandwidth-percent  Set EIGRP bandwidth limit`
`  broadcast-address  Set the broadcast address of an interface`
`  cgmp  Enable/disable CGMP`
`  directed-broadcast  Enable forwarding of directed broadcasts`
`  dvmrp  DVMRP interface commands`
`  hello-interval  Configures IP-EIGRP hello interval`
`  helper-address  Specify a destination address for UDP broadcasts`
`  hold-time  Configures IP-EIGRP hold time`
`  .`
`  .`
`  .`
`Router(config-if)# ip`
	Enter the command that you want to configure for the interface. This example uses the **IP** command.
	Enter **?** to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands.

`Router(config-if)# ip address ?`
`  A.B.C.D  IP address`
`  negotiated  IP Address negotiated over PPP`
`Router(config-if)# ip address`
	Enter the command that you want to configure for the interface. This example uses the **IP address** command.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP address or the **negotiated** keyword.
	A carriage return (<cr>) is not displayed. Therefore, you must enter additional keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 ?`
`  A.B.C.D  IP subnet mask`
`Router(config-if)# ip address 172.16.0.1`
	Enter the keyword or argument that you want to use. This example uses the 172.16.0.1 IP address.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask.
	<cr> is not displayed. Therefore, you must enter additional keywords or arguments to complete the command.

Command	Comment
Router(config-if)# ip address 172.16.0.1 255.255.255.0 ? secondary Make this IP address a secondary address <cr> Router(config-if)# ip address 172.16.0.1 255.255.255.0	Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask. Enter **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword, or you can press **Enter**. <cr> is displayed. Press **Enter** to complete the command, or enter another keyword.
Router(config-if)# ip address 172.16.0.1 255.255.255.0 Router(config-if)#	Press **Enter** to complete the command.

Software advisor

Software Advisor is a Cisco tool that:

- checks if a feature is supported in a Cisco IOS XE release
- locates the software document for that feature, and
- verifies the minimum software requirements of Cisco IOS XE software with the hardware installed on your router.

Software advisor access requirements

You must be a registered user on Cisco.com to access this tool. Cisco maintains the Software Advisor tool. See Tools and Resources.

Use software release notes

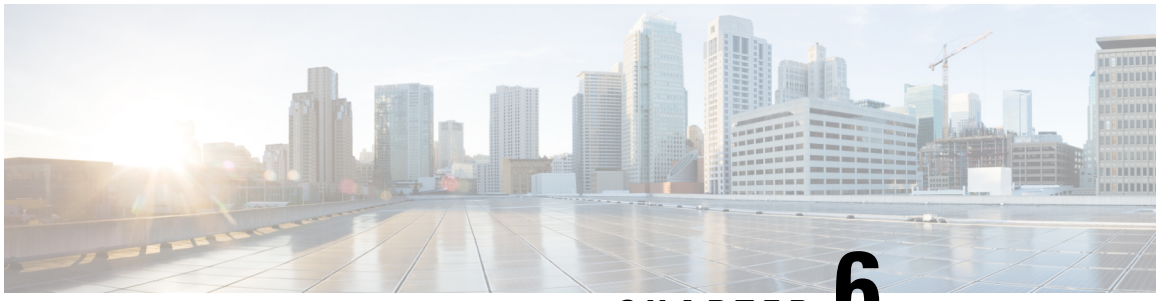
Software release notes provide essential information about the current release. This reference lists the types of information available in release notes and explains how to access cumulative feature information from previous releases.

Release notes contain information about these topics:

- Product overview
- Open and resolved severity 1 and 2 caveats
- Software image names
- New features
- Known limitations

Release notes are intended to be release-specific for the most current release, and the information provided in these documents may not be cumulative in providing information about features that first appeared in

previous releases. For cumulative feature information, refer to the Cisco Feature Navigator at: http://www.cisco.com/go/cfn/.



CHAPTER 6

Basic Router Configuration

This chapter contains the following sections:

- ESR6300 interface naming, on page 69
- Basic Configuration, on page 70
- Configuring Global Parameters, on page 75
- Serial port support, on page 76
- Rawsocket Keepalive Configuration CLI, on page 76
- Configuring the Gigabit Ethernet Interface, on page 77
- Configuring a Loopback Interface, on page 78
- Enabling Cisco Discovery Protocol, on page 79
- Configuring Command-Line Access, on page 80
- Configuring Static Routes, on page 81
- Configuring Dynamic Routes, on page 83
- Modular QoS (MQC), on page 85

ESR6300 interface naming

The supported hardware interfaces and their naming conventions are in this table:

Hardware Interface	IOS-XE Naming Convention
Gigabit Ethernet combo port WAN/Layer3	gigabitEthernet 0/0/0 gigabitEthernet 0/0/1
Gigabit Ethernet LAN/Layer 2 ports	gigabitEthernet 0/1/0 gigabitEthernet 0/1/1 gigabitEthernet 0/1/2 gigabitEthernet 0/1/3
Console Port	Line console 0
USB Port	usbflash0: (IOS and rommon)

Basic Configuration

The basic configuration is a result of the entries you made during the initial configuration dialog. This means the router has at least one interface set with an IP address to be reachable, either through WebUI or to allow the PnP process to work. Use the **show running-config** command to view the initial configuration, as shown in the following example:

```
Router# show running-config
Building configuration...


Current configuration : 13001 bytes
!
! Last configuration change at 12:17:45 UTC Wed Sep 25 2019
!
version 17.1
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
platform hardware throughput level 2G
no platform punt-keepalive disable-kernel-core
!
hostname Router
!
boot-start-marker
boot system bootflash:c6300-universalk9.2019-09-23_04.57_gpollepa.SSA.bin
boot-end-marker
!
!
no logging monitor
!
aaa new-model
--More--
*Sep 25 23:53:59.524: %PARSER-5-CFGLOG_LOGGEDCMD: User:console logged command:!ex! e
!
aaa authentication enable default none
aaa authorization exec default local
!
aaa session-id common
clock timezone UTC -8 0
call-home
! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
! the email address configured in Cisco Smart License Portal will be used as contact email
 address to send SCH notifications.
contact-email-addr sch-smart-licensing@cisco.com
profile "CiscoTAC-1"
active
destination transport-method http
!
login on-success log
ipv6 unicast-routing
!
subscriber templating
!
multilink bundle-name authenticated
!
flow exporter 10.10.10.11
destination 10.10.10.11
!
crypto pki trustpoint SLA-TrustPoint
```

```
revocation-check crl
!
crypto pki trustpoint DNAC-CA
enrollment mode ra
enrollment terminal
usage ssl-client
revocation-check crl none
!
crypto pki trustpoint TP-self-signed-2633875772
enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-2633875772
revocation-check none
rsakeypair TP-self-signed-2633875772
!
!
crypto pki certificate chain SLA-TrustPoint
certificate ca 01
30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
D697DF7F 28
quit
crypto pki certificate chain DNAC-CA
crypto pki certificate chain TP-self-signed-2633875772
certificate self-signed 01
30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
69666963 6174652D 32363333 38373537 3732301E 170D3139 30393235 30303034
35305A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 36333338
37353737 32308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
0A028201 0100E848 4E2F3B31 0FA018F3 32D29A16 D8FDD1E1 14820D00 1D9A0AEF
994283C8 938803DA 3344AF4E BB4E94CD A8FDD1B8 AE35DC14 97BB8EC6 DC475C5F
438D80A9 6EC125BE 77246D7C 7D3F15DA CA340EDE 238595C5 CE70A762 315934B4
CA08037A 83E31EFB 5D070986 3C2553C0 C5B63A45 3A1FE935 EF0F74E2 2D5AC8C8
F21408A8 6AE6C799 D0F882D1 A2CA684D 29DA01A3 A31527F1 C613FF3F 547CEC2F
82ADFC5E 9461AA9A 546448AC BB3D82B6 449D3BC0 C844C92B 9D951423 7869250E
936E1147 AF8C5367 D34ECCA7 833AF8D1 CAC6F0CD 8A41BDD3 AE68CBD7 FA0794FF
D984763C 33E7F772 13A0E35B C9FA823E F6262FA1 269407B1 68BE318D EB3E13E2
2C32B3DF FC9D0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 14F5FA3B 95F59783 28BF5857 1EF8C95B CC6F758C
98301D06 03551D0E 04160414 F5FA3B95 F5978328 BF58571E F8C95BCC 6F758C98
```

```
300D0609 2A864886 F70D0101 05050003 82010100 31F5A7B6 DA464923 BFB45830
24FB2D90 D044B831 EDF45EB8 7671C54B 9F309AC9 1BE168EB C4AE8142 CE939520
4633531E 8DFFA22A A46373F5 9AEFFBDC A33DC696 C8606AEB 0771B923 E3731049
CE802EE9 825A93AD EA304F4B DE495E02 25142F57 4B50CBF7 08480E62 BF9587E4
0303DB1E 5B524CA9 0AAFF669 7319074B CE560D90 409F0976 60DE6299 12F360EC
36D50FB1 8471FF91 5C613CDE 836C1CD0 DB876E34 60BA6FF8 8DBE4F76 180F0A36
DAEB023C C3672AB6 F3CA2792 6585700A E9B3F173 08D02311 8FBECDD5 0B4D459E
5352D388 7414DED9 964AB488 68F6B3D7 0EB39A46 2351DBCD 8FD97D24 DFA4C9E1
02A312DF 31E49F1F F10F9D39 D9A7B12A 62F7875A
quit
!
no license feature hseck9
license udi pid ESR-6300-NCP-K9 sn FOC23032UUN
license boot level network-advantage addon dna-advantage
archive
log config
logging enable
logging size 1000
notify syslog contenttype plaintext
path bootflash:saved-config
maximum 5
memory free low-watermark processor 49965
!
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
username cisco privilege 15 password 0 cisco
!
redundancy
mode none
!
vlan internal allocation policy ascending
!
lldp run
!
bridge irb
!
interface VirtualPortGroup0
ip address 192.168.0.1 255.255.255.0
ip nat inside
ipv6 address autoconfig
ipv6 enable
no mop enabled
no mop sysid
!
interface GigabitEthernet0/0/0
ip address 172.27.168.161 255.255.255.128
media-type rj45
negotiation auto
!
interface GigabitEthernet0/0/1
ip address 15.0.0.1 255.255.255.0
ip nat inside
shutdown
media-type sfp
negotiation auto
ipv6 enable
!
interface GigabitEthernet0/1/0
shutdown
!
interface GigabitEthernet0/1/1
shutdown
```

```
!
interface GigabitEthernet0/1/2
shutdown
!
interface GigabitEthernet0/1/3
shutdown
!
interface Vlan1
no ip address
!
interface vmi1
no ip address
!
iox (Note: Some protocols appear in the configuration although they are not supported)
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/0/0 172.27.168.129
ip ssh rsa keypair-name sshkeys
ip ssh version 2
!
snmp-server group mac v3 priv read mac write mac notify maca
snmp-server group fips v3 priv read fips write fips notify fips
snmp-server group cgnms v3 priv
snmp-server group cg-nms-administrator v3 priv
snmp-server view mac mib-2 included
snmp-server view mac system included
snmp-server view mac sysUpTime included
snmp-server view fips mib-2 included
snmp-server view fips system included
snmp-server view fips sysUpTime included
snmp-server community public RO R0
snmp-server community private RW
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps pfr
snmp-server enable traps flowmon
snmp-server enable traps ds1
snmp-server enable traps entity-perf throughput-notif
snmp-server enable traps ds3
snmp-server enable traps call-home message-send-fail server-fail
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change shamlink interface
snmp-server enable traps ospf cisco-specific state-change shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps casa
snmp-server enable traps xgcp
snmp-server enable traps license
snmp-server enable traps smart-license
snmp-server enable traps cef resource-failure peer-state-change peer-fib-state-change
inconsistency
snmp-server enable traps memory bufferpeak
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps config-ctid
```

```
snmp-server enable traps dhcp
snmp-server enable traps dsp card-status
snmp-server enable traps dsp oper-state
snmp-server enable traps dsp video-usage
snmp-server enable traps dsp video-out-of-resource
snmp-server enable traps fru-ctrl
snmp-server enable traps entity
snmp-server enable traps event-manager
snmp-server enable traps frame-relay multilink bundle-mismatch
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps hsrp
snmp-server enable traps pimstdmib neighbor-loss invalid-register invalid-join-prune
rp-mapping-change interface-election
snmp-server enable traps ipmulticast
snmp-server enable traps isis
snmp-server enable traps ip local pool
snmp-server enable traps msdp
snmp-server enable traps mvpn
snmp-server enable traps ospfv3 state-change
snmp-server enable traps ospfv3 errors
snmp-server enable traps pim neighbor-change rp-mapping-change invalid-pim-message
snmp-server enable traps pppoe
snmp-server enable traps cpu threshold
snmp-server enable traps rsvp
snmp-server enable traps syslog
snmp-server enable traps l2tun session
snmp-server enable traps l2tun pseudowire status
snmp-server enable traps aaa_server
snmp-server enable traps atm subif
snmp-server enable traps pki
snmp-server enable traps ethernet evc status create delete
snmp-server enable traps ether-oam
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
snmp-server enable traps entity-state
snmp-server enable traps vdsl2line
snmp-server enable traps entity-sensor
snmp-server enable traps flash insertion removal lowspace
snmp-server enable traps srp
snmp-server enable traps nhrp nhs
snmp-server enable traps nhrp nhc
snmp-server enable traps nhrp nhp
snmp-server enable traps nhrp quota-exceeded
snmp-server enable traps otn
snmp-server enable traps pw vc
snmp-server enable traps ipsla
snmp-server enable traps sonet
snmp-server enable traps dlsw
snmp-server enable traps resource-policy
snmp-server enable traps lisp
snmp-server enable traps stpx inconsistency root-inconsistency loop-inconsistency
snmp-server enable traps dial
snmp-server enable traps sbc adj-status
snmp-server enable traps sbc blacklist
snmp-server enable traps sbc congestion-alarm
snmp-server enable traps c3g
snmp-server enable traps LTE
snmp-server enable traps vtp
snmp-server enable traps vlancreate
snmp-server enable traps ike policy add
snmp-server enable traps ike policy delete
snmp-server enable traps ipsec cryptomap add
snmp-server enable traps ipsec cryptomap delete
```

```
snmp-server enable traps ipsec cryptomap attach
snmp-server enable traps ipsec cryptomap detach
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps gdoi gm-start-registration
snmp-server enable traps gdoi gm-registration-complete
snmp-server enable traps gdoi gm-re-register
snmp-server enable traps gdoi gm-rekey-rcvd
snmp-server enable traps gdoi gm-rekey-fail
snmp-server enable traps gdoi ks-rekey-pushed
snmp-server enable traps gdoi gm-incomplete-cfg
snmp-server enable traps gdoi ks-no-rsa-keys
snmp-server enable traps gdoi ks-new-registration
snmp-server enable traps gdoi ks-reg-complete
snmp-server enable traps gdoi ks-role-change
snmp-server enable traps gdoi ks-gm-deleted
snmp-server enable traps gdoi ks-peer-reachable
snmp-server enable traps gdoi ks-peer-unreachable
snmp-server enable traps firewall serverstatus
snmp-server enable traps bulkstat collection transfer
snmp-server enable traps alarms informational
snmp-server enable traps vrfmib vrf-up vrf-down vnet-trunk-up vnet-trunk-down
!
control-plane
!
mgcp behavior rsip-range tgcp-only
mgcp behavior comedia-role none
mgcp behavior comedia-check-media-src disable
mgcp behavior comedia-sdp-force disable
!
mgcp profile default
!
line con 0
stopbits 1
speed 115200
line vty 0 4
transport input all
transport output all
line vty 5 15
transport input all
transport output all
!
app-hosting system-resources
end
```

Configuring Global Parameters

To configure global parameters for your router, follow these steps.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode when using the console port.

	Command or Action	Purpose

	`Router> enable`
	`Router# configure terminal`
	`Router(config)#`
		Use the following to connect to the router with a remote terminal:
		`telnet router-name or address`
		`Login: login-id`
		`Password: *********`
		`Router> enable`

Step 2	**hostname** *name* **Example:** `Router(config)# hostname ESR6300`	Specifies the name for the router.
Step 3	**enable password** *password* **Example:** `Router(config)# enable password cr1ny5ho`	Specifies a password to prevent unauthorized access to the router. **Note** In this form of the command, password is not encrypted. To encrypt the password use enable secret password as noted in the previously mentioned Device Hardening Guide.

Serial port support

Serial port support is a set of protocol capabilities that:

- enables SCADA Gateway functionality (IEC10x and DNP3),
- provides Raw Socket (TCP and UDP) support,
- includes Line Relay and Reverse Telnet capabilities, and
- brings the ESR6300 into feature compatibility with the IR1101.

Configuration compatibility

All of the configuration and show commands will be the same as are available on the IR1101 platform.

For detailed configuration information, see Cisco Catalyst IR1101 Rugged Series Router Software Configuration Guide.

Rawsocket Keepalive Configuration CLI

Rawsocket keepalive for async interfaces is a feature that existed in classic IOS platforms. As part of 17.10.1a, the feature will be extended to IOS-XE based platforms. A new CLI with the following syntax will be added under rawsocket.

`Router(config-line)#raw-socket tcp keepalive` *interval*

CLI Changes

On IOS-XE platforms starting from 17.10.1a, there is a CLI correction and an additional CLI was added as part of raw-socket.

The correction is for the **raw-socket idle timeout** command. There is now an option to configure the timeout based on minutes and seconds, whereas the previous configuration used only minutes.

```
Router(config-line)# raw-socket tcp idle-timeout [0-1440] [<0-59> | cr]
```

The additional CLI is for clearing the raw-socket TCP clients. The command syntax is **clear raw-socket line** *[1-145|tty|x/y/z]* for example:

```
Router# clear raw-socket line 0/2/0
```



Note When initiating clear raw-socket line, raw-socket sessions will be cleared for raw-socket clients from the **show raw-socket tcp sessions** command. Connections will be re-established after a TCP hand-shake, which can be done by doing shut/no shut on TCP connection interface.

Configuring the Gigabit Ethernet Interface

The default configuration for the Gigabit Ethernet Interface (GI0/0/0 and GI0/0/1) on the ESR6300 are only Layer 3 (L3). The Gigabit Ethernet Interface on the ESR6300 is a combo port, which means it is a RJ45+SFP connector. These interfaces are combo ports, which means you can connect either through the RJ45 or SFP port by configuring the "media-type" to either auto-select, rj45 or sfp

To manually define the Gigabit Ethernet interface, follow these steps, beginning from global configuration mode.

Procedure

	Command or Action	Purpose
Step 1	**interface GigabitEthernet** *slot/bay/port* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Enters the configuration mode for an interface on the router.
Step 2	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.12.2 255.255.255.0`	Sets the IP address and subnet mask for the specified interface. Use this Step if you are configuring an IPv4 address.
Step 3	**ipv6 address** *ipv6-address/prefix* **Example:** `Router(config-if)# ipv6 address 2001.db8::ffff:1/128`	Sets the IPv6 address and prefix for the specified interface. Use this step instead of Step 2, if you are configuring an IPv6 address. IPv6 unicast-routing needs to be set-up as well, see further information in the **IPv6 Addressing and**

	Command or Action	Purpose
		Basic Connectivity Configuration Guide located here: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_basic/configuration/xe-16-10/ip6b-xe-16-10-book/read-me-first.html
Step 4	**ipv6 unicast-routing** **Example:** `Router (config)# ipv6 unicast-routing`	Enables forwarding of IPv6 unicast data packets.
Step 5	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface and changes its state from administratively down to administratively up.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits the configuration mode of interface and returns to the global configuration mode.

Configuring a Loopback Interface

Before you begin

The loopback interface acts as a placeholder for the static IP address and provides default routing information.

To configure a loopback interface, follow these steps.

Procedure

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:** `Router(config)# interface Loopback 0`	Enters configuration mode on the loopback interface.
Step 2	(Option 1) **ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.108.1.1 255.255.255.0`	Sets the IP address and subnet mask on the loopback interface. (If you are configuring an IPv6 address, use the **ipv6 address** *ipv6-address/prefix* command described below.
Step 3	(Option 2) **ipv6 address** *ipv6-address/prefix* **Example:**	Sets the IPv6 address and prefix on the loopback interface.

	Command or Action	Purpose
	`Router(config-if)# ipv6 address 2001:db8::ffff:1/128`	
Step 4	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode for the loopback interface and returns to global configuration mode.

Example

Verifying Loopback Interface Configuration

Enter the **show interface loopback** command. You should see an output similar to the following example:

```
Router# show interface loopback 0
Loopback0 is up, line protocol is up
  Hardware is Loopback
  Internet address is 192.0.2.0/16
  MTU 1514 bytes, BW 8000000 Kbit, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, loopback not set
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Alternatively, use the **ping** command to verify the loopback interface, as shown in the following example:

```
Router# ping 192.0.2.0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.0, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Enabling Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) is enabled by default on the router. It may be disabled if needed for security purposes.

For more information on using CDP, see Cisco Discovery Protocol Configuration Guide, Cisco IOS XE Release 3S.

Configuring Command-Line Access

To configure parameters to control access to the router, follow these steps.



Note Transport input must be set as explained in the previous Telnet and SSH sections of the guide.

Procedure

	Command or Action	Purpose
Step 1	**line** [**aux** | **console** | **tty** | **vty**] *line-number* **Example:** `Router(config)# line console 0`	Enters line configuration mode, and specifies the type of line. The example provided here specifies a console terminal for access.
Step 2	**password** *password* **Example:** `Router(config-line)# password 5dr4Hepw3`	Specifies a unique password for the console terminal line.
Step 3	**login** **Example:** `Router(config-line)# login`	Enables password checking at terminal session login.

Step 4	**exec-timeout** *minutes* [*seconds*]
	Example:
	`Router(config-line)# exec-timeout 5 30`
	`Router(config-line)#`
		Sets the interval during which the EXEC command interpreter waits until user input is detected. The default is 10 minutes. Optionally, adds seconds to the interval value.
		The example provided here shows a timeout of 5 minutes and 30 seconds. Entering a timeout of **0 0** specifies never to time out.

Step 5	**exit** **Example:** `Router(config-line)# exit`	Exits line configuration mode to re-enter global configuration mode.

Step 6	**line** [**aux** | **console** | **tty** | **vty**] *line-number*
	Example:
	`Router(config)# line vty 0 4`
	`Router(config-line)#`
		Specifies a virtual terminal for remote console access.

	Command or Action	Purpose
Step 7	**password** *password* **Example:** `Router(config-line)# password aldf2ad1`	Specifies a unique password for the virtual terminal line.
Step 8	**login** **Example:** `Router(config-line)# login`	Enables password checking at the virtual terminal session login.
Step 9	**end** **Example:** `Router(config-line)# end`	Exits line configuration mode, and returns to privileged EXEC mode.

Configuring Static Routes

Static routes provide fixed routing paths through the network. They are manually configured on the router. If the network topology changes, the static route must be updated with a new route. Static routes are private routes unless they are redistributed by a routing protocol.

To configure static routes, follow these steps.

Procedure

	Command or Action	Purpose
Step 1	(Option 1) **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} **Example:** `Router(config)# ip route 192.10.2.3 255.255.0.0 10.10.10.2`	Specifies a static route for the IP packets. (If you are configuring an IPv6 address, use the **ipv6 route** command described below.)
Step 2	(Option 2) **ipv6 route** *prefix/mask* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} **Example:** `Router(config)# ipv6 route 2001:db8:2::/64 2001:db8:3::0`	Specifies a static route for the IP packets. See additional information for IPv6 here: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_basic/configuration/xe-16-10/ip6b-xe-16-10-book/read-me-first.html

	Command or Action	Purpose
Step 3	**end** **Example:** `Router(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

In the following configuration example, the static route sends out all IP packets with a destination IP address of 192.168.1.0 and a subnet mask of 255.255.255.0 on the Gigabit Ethernet interface to another device with an IP address of 10.10.10.2. Specifically, the packets are sent to the configured PVC.

You do not have to enter the command marked **default**. This command appears automatically in the configuration file generated when you use the **running-config** command.

```
!
ip classless (default)
ip route 2001:db8:2::/64 2001:db8:3::0
```

Verifying Configuration

To verify that you have configured static routing correctly, enter the **show ip route** command (or **show ipv6 route** command) and look for static routes marked with the letter S.

When you use an IPv4 address, you should see verification output similar to the following:

```
Router# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.108.1.0 is directly connected, Loopback0
S*   0.0.0.0/0 is directly connected, FastEthernet0
```

When you use an IPv6 address, you should see verification output similar to the following:

```
Router# show ipv6 route
IPv6 Routing Table - default - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE -
Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       ls - LISP site, ld - LISP dyn-EID, a - Application

C   2001:DB8:3::/64 [0/0]
       via GigabitEthernet0/0/2, directly connected
S   2001:DB8:2::/64 [1/0]
       via 2001:DB8:3::1
```

Configuring Dynamic Routes

In dynamic routing, the network protocol adjusts the path automatically, based on network traffic or topology. Changes in dynamic routes are shared with other routers in the network.

All of the Cisco IOS-XE configuration guides can be found here: https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xe-gibralter-16-10-1/model.html#ConfigurationGuides

Configuring Routing Information Protocol

To configure the RIP on a router, follow these steps.

Procedure

	Command or Action	Purpose
Step 1	**router rip** **Example:** `Router(config)# router rip`	Enters router configuration mode, and enables RIP on the router.
Step 2	**version** {**1** | **2**} **Example:** `Router(config-router)# version 2`	Specifies use of RIP version 1 or 2.

Step 3	**network** *ip-address*
	Example:
	`Router(config-router)# network 192.168.1.1`
	`Router(config-router)# network 10.10.7.1`
		Specifies a list of networks on which RIP is to be applied, using the address of the network of each directly connected network.

Step 4	**no auto-summary** **Example:** `Router(config-router)# no auto-summary`	Disables automatic summarization of subnet routes into network-level routes. This allows subprefix routing information to pass across classful network boundaries.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode, and enters privileged EXEC mode.

Example

Verifying Configuration

To verify that you have configured RIP correctly, enter the **show ip route** command and look for RIP routes marked with the letter R. You should see an output similar to the one shown in the following example:

```
Router# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.108.1.0 is directly connected, Loopback0
R    3.0.0.0/8 [120/1] via 2.2.2.1, 00:00:02, Ethernet0/0/0
```

Configuring Enhanced Interior Gateway Routing Protocol

The Enhanced Interior Gateway Routing Protocol (EIGRP) is an enhanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco. The convergence properties and the operating efficiency of EIGRP have improved substantially over IGRP, and IGRP is now obsolete.

The convergence technology of EIGRP is based on an algorithm called the Diffusing Update Algorithm (DUAL). The algorithm guarantees loop-free operation at every instant throughout a route computation and allows all devices involved in a topology change to synchronize. Devices that are not affected by topology changes are not involved in recomputations

Details on configuring Enhanced Interior Gateway Routing Protocol (EIGRP), are found in the following guide: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_eigrp/configuration/xe-16-10/ire-xe-16-10-book/ire-enhanced-igrp.html

Configuring Open Shortest Path First (OSPF)

OSPF is an Interior Gateway Protocol (IGP) developed by the OSPF working group of the Internet Engineering Task Force (IETF). OSPF was designed expressly for IP networks and it supports IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets.

Cisco supports RFC 1253, *OSPF Version 2 Management Information Base*, August 1991. The OSPF MIB defines an IP routing protocol that provides management information related to OSPF and is supported by Cisco routers.

Details on configuring OSPF are found in the following guide:https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/xe-16-12/iro-xe-16-12-book.html

Configuring Integrated Intermediate System-to-Intermediate System (IS-IS) Routing Protocol

IS-IS is a link-state Interior Gateway Protocol (IGP). Link-state protocols are characterized by the propagation of the information required to build a complete network connectivity map on each participating device. That map is then used to calculate the shortest path to destinations.

The IS-IS protocol was developed in the late 1980s by Digital Equipment Corporation (DEC) and was standardized by the International Standards Organization (ISO) in ISO/IEC 10589. The current version of this standard is ISO/IEC 10589:2002.

ISO/IEC 10589 defines support for the ISO Connectionless Network Protocol (CLNP) as defined in ISO 8473. However, the protocol was designed to be extensible to other network protocols. RFC 1195 defined IS-IS support for IP, and additional IETF extensions have defined IS-IS support for IPv6. Integration of support for multiple network layer protocols has led to the term Integrated IS-IS. The Cisco IOS IS-IS implementation supports CLNP, IPv4, and IPv6. This module and its related modules use the term IS-IS to refer to the Integrated IS-IS that is implemented by Cisco IOS software.

For details on configuring IS-IS, see the following guide:

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_isis/configuration/xe-16-12/irs-xe-16-12-book.html

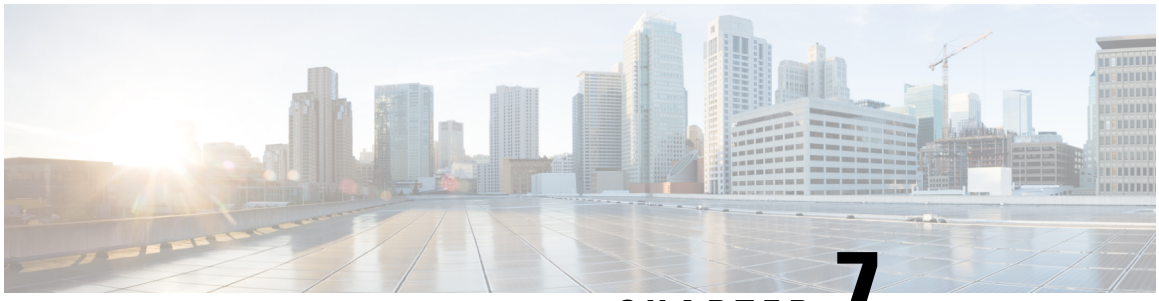
Modular QoS (MQC)

Modular QoS CLI (MQC) is a standardized approach that

- enables all QoS features to be configured on the IoT Integrated Services Router
- provides a unified command-line interface for QoS configuration, and
- operates across Cisco routing and switching platforms.

Configuration guidance

For more information, see Follow the procedures that are in the QoS Modular QoS Command-Line Interface Configuration Guide, Cisco IOS XE 17 guide.



CHAPTER 7

Cisco IOS XE installation methods

Provides an overview of the supported procedures for installing Cisco IOS XE software on network devices.

- Bundle mode versus install mode, on page 87
- Installing software using install commands, on page 88
- Restrictions for installing the software using install commands, on page 88
- Install mode support, on page 88
- Information about installing the software using install commands, on page 89
- Configuration examples, on page 96
- Troubleshoot software installation using install commands, on page 103

Bundle mode versus install mode

Bundle mode and install mode are BOOT methods that:

- determine how Cisco IOS XE software loads on IoT routers
- use different file formats and storage locations for the operating system, and
- provide different levels of control over the software installation process.

BOOT mode characteristics

Bundle mode uses a single compressed image with the typical naming convention <product>-universalk9.<release>.SPA.bin. This mode provides a consolidated BOOT process, using local (hard disk, flash) or remote (TFTP) .bin image. Booting via a .bin image means that the router would first have to uncompress the image before booting from it. This led to a longer period of time for the router to BOOT.

To upgrade the router to a new version of IOS XE, you would point the "BOOT system" to a new software image. This method is well known and details are available in your products configuration guide.

Starting with IOS XE release 17.9.1, install mode has been added to the IoT routers. Install mode uses packages loaded into bootflash, which are read by a packages.conf file. This method provides more control over the software installation process.

Install mode requires more room in bootflash: for the files. The packages are slightly larger than the .bin images, and they vary per product in size.

Note If the `BOOT` variable points to `flash:packages.conf`, the device is in install mode. If it points to a `.bin` file, it is in bundle mode.

Installing software using install commands

Installing software using install commands is a software management method that enables users to boot the platform and upgrade or downgrade to Cisco IOS XE software versions using a set of **install** commands on Cisco IoT routers shipped in install mode by default from Cisco IOS XE 17.9.1.

Restrictions for installing the software using install commands

Be aware of these restrictions when installing software using install commands:

- Install mode requires a reboot of the system.
- SMU installation was supported in both bundle boot and install mode. From Cisco IOS XE Release 17.9.x, SMU installation will be stopped if the router is booted up in bundle mode. If the router is booted up in install mode, SMU installation will keep working as it is in previous releases.

Install mode support

This table describes the differences between Bundle mode and Install mode.

Cisco IOS XE running on IoT routers has typically made use of the Bundle boot mode. Bundle boot mode is also known as Consolidated boot, and uses a single compressed image. The typical naming convention is <product>-universalk9.<release>.SPA.bin.

This mode provides a consolidated boot process, using local (hard disk, flash) or remote (TFTP) .bin image. Booting via a .bin image means that the router would first have to uncompress the image before booting from it. This led to a longer period of time for the router to boot.

To upgrade the router to a new version of IOS XE, you would point the "boot system" to a new software image. This method is well known and details are available in your products configuration guide.

Starting with IOS XE release 17.9.1, a new boot mode called Install mode has been added to the IoT routers. Install mode uses packages loaded into bootflash, which are read by a packages.conf file. This method provides more control over the software installation process.

Note SMU installation was supported in both bundle boot and install mode. From Cisco IOS XE Release 17.9.x, SMU installation will be stopped if the router is booted up in bundle mode. If the router is booted up in install mode, SMU installation will keep working as it is in previous releases.

Table 6: Bundle mode vs install mode

Bundle Mode	Install Mode
This mode provides a consolidated boot process, using local (hard disk, flash) or remote (TFTP) .bin image.	This mode uses the local (bootflash) packages.conf file for the boot process.
This mode uses a single .bin file.	.bin file is replaced with expanded .pkg files in this mode.
CLI: `Router(config)#boot system bootflash:<filename>`	CLI: `#install add file bootflash: [activate commit]`
To upgrade in this mode, point the boot system to the new image.	To upgrade in this mode, use the **install** commands.
Image Auto-Upgrade: When a new Field-Replaceable Unit (FRU) is inserted in a modular chassis, manual intervention is required to get the new FRU running with the same version as the active FRUs.	Image Auto-Upgrade: When a new FRU is inserted in a modular chassis, the joining FRU is auto-upgraded to the image version in sync with the active FRUs.
Rollback: Rollback to the previous image with multiple Software Maintenance Updates (SMUs) may require multiple reloads.	Rollback: Enables rollback to an earlier version of Cisco IOS XE software, including multiple patches in single reload.

For additional information, please see Cisco IOS XE Installation Methods.

Information about installing the software using install commands

From the Cisco IOS XE 17.9.1 release, IoT routers will be shipped in install mode instead of bundle mode. So any new router from the factory will boot up in install mode.

Existing installations using previous releases of IOS XE have the option to continue to use their device in Bundle mode if they wish to. Or they can convert their device to Install mode.

Install mode is applicable to both autonomous mode and controller mode.

A new release can be installed in Install mode using vManage.

Table 7: Bundle mode vs install mode

Bundle Mode	Install Mode
This mode provides a consolidated boot process, using local (hard disk, flash) or remote (TFTP) .bin image.	This mode uses the local (bootflash) packages.conf file for the boot process.
This mode uses a single .bin file.	.bin file is replaced with expanded .pkg files in this mode.

Bundle Mode	Install Mode
CLI: `Router(config)#boot system bootflash:<filename>`	CLI: `#install add file bootflash: [activate commit]`
To upgrade in this mode, point the boot system to the new image.	To upgrade in this mode, use the **install** commands.
Image Auto-Upgrade: When a new Field-Replaceable Unit (FRU) is inserted in a modular chassis, manual intervention is required to get the new FRU running with the same version as the active FRUs.	Image Auto-Upgrade: When a new FRU is inserted in a modular chassis, the joining FRU is auto-upgraded to the image version in sync with the active FRUs.
Rollback: Rollback to the previous image with multiple Software Maintenance Updates (SMUs) may require multiple reloads.	Rollback: Enables rollback to an earlier version of Cisco IOS XE software, including multiple patches in single reload.

Installing software using install mode commands

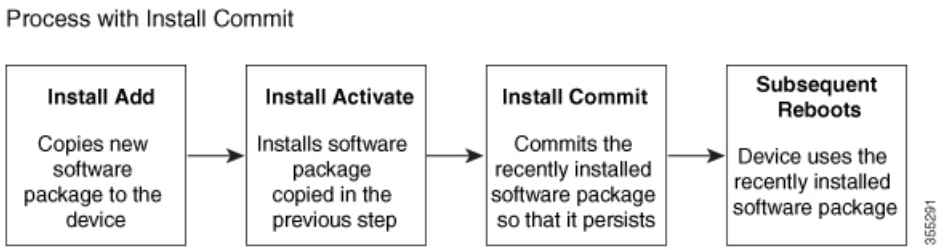
Summary

The key components involved in the install mode process are:

- **install add**: Copies software package from local or remote location to platform and validates the image file
- **install activate**: Performs validations and provisions packages previously added, triggering system reload
- **install commit**: Confirms packages previously activated and makes updates persistent over reloads

The install mode process flow comprises three commands to perform installation and upgrade of software on platforms. Installing an update replaces any previously installed software image, with only one image installed at any time.

Workflow



These stages describe how the install mode process works:

1. The system copies and validates the software package using the **install add** command.
    - Validates the file–checksum, platform compatibility checks, and so on
    - Extracts individual components of the package into subpackages and packages.conf

- Copies the image into the local inventory and makes it available for the next steps

The **install add** command copies the software package from a local or remote location to the platform. The command extracts individual components of the .package file into subpackages and packages.conf files. It also validates the file to ensure that the image file is specific to the platform on which it is being installed.

The location of the software package can be in several places, as shown in the output of this command:

```
Router# install add file ?
bootflash: Package name
crashinfo: Package name
flash: Package name
ftp: Package name
http: Package name
https: Package name
pram: Package name
rcp: Package name
scp: Package name
sftp: Package name
tftp: Package name
webui: Package name
```

2. The system activates the package using the **install activate** command and triggers a reload.
   - Use the **show install summary** command to see which image is inactive. This image will get activated
   - System reloads on executing this command. Confirm if you want to proceed with the activation. Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts

   The **install activate** command performs the required validations and provisions the packages previously added using the **install add** command. It also triggers a system reload.

   The auto-abort timer starts automatically with a default value of 120 minutes. If the **install commit** command is not executed within the time provided, the activation process is terminated, and the system returns to the last-committed state.

3. The system commits the activated package using the **install commit** command to make changes persistent.
   - Use the **show install summary** command to see which image is uncommitted. This image will get committed

   The **install commit** command confirms the packages previously activated using the **install activate** command, and makes the updates persistent over reloads.

Boot the platform in install mode

You can install, activate, and commit a software package using a single command (one-step install) or multiple separate commands (three-step install).

If the platform is working in bundle mode, the one-step install procedure must be used to initially convert the platform from bundle mode to install mode. Subsequent installs and upgrades on the platform can be done with either one-step or three-step variants.

You can see how your device is set up to boot by using the **show romvar** and **show bootvar** commands.

```
Router# show romvar
ROMMON variables:
```

```
PS1 = rommon ! >
CM = c6300
DEVICE_MANAGED_MODE = autonomous
LICENSE_SUITE =
RET_2_RTS =
THRPUT = 250
BOOT = flash:packages.conf,12;
LICENSE_BOOT_LEVEL = network-advantage,all:c6300;
BSI = 0
RET_2_RCALTS =
RANDOM_NUM = 212626522
Router#

Router# show bootvar
BOOT variable = flash:packages.conf,12;
CONFIG_FILE variable does not exist
BOOTLDR variable does not exist
Configuration register is 0x2102

Standby not ready to show bootvar

Router#
```

Convert from bundle mode to install mode or upgrade the platform using one-step installation

Use the one-step install procedure to convert a platform running in bundle boot mode to install mode. After the command is executed, the platform reboots in install boot mode. Later, you can also use the one-step install procedure to upgrade the platform.

This procedure uses the `install add file activate commit` command in privileged EXEC mode to install a software package and to upgrade the platform to a new version.

Note

- All the CLI actions (for example, add and activate) are executed.
- The configuration save prompt appears if an unsaved configuration is detected.
- The reload prompt appears after the second step in this workflow. Use the `prompt-level none` keyword to automatically ignore the confirmation prompts.
- If the prompt-level is set to None and there is an unsaved configuration, the install fails. You must save the configuration before reissuing the command.

Follow these steps to convert from bundle mode to install mode, or to upgrade the platform using the one-step installation.

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 Use the **install add file** *location*: *filename* [**activate commit**] command to copy the software install package to the platform and extract its components, validate platform and image compatibility, activate the package, and commit it to make it persistent across reloads.

Example:

```
Router# install add file bootflash:<router_image>.SSA.bin activate commit
```

Note

- You can copy the software install package from a local or remote location (through FTP, HTTP, HTTPS, or TFTP).
- This command extracts the individual components of the .package file into subpackages and packages.conf files.
- The platform reloads after this command is run.

Step 3 Use the **exit** command to exit privileged EXEC mode and return to user EXEC mode.

Example:

```
Router# exit
```

The platform converts from bundle boot mode to install boot mode and reboots. The platform is now ready for future upgrades using the one-step install procedure.

Three-step installation

This procedure provides more flexibility and control during software installation by using the individual `install add`, `install activate`, and `install commit` commands to install a software package and upgrade the platform to a new version.

Note

- You can use the three-step installation procedure only after the platform is in install mode.
- All the CLI actions (for example, add and activate) are executed.
- The configuration save prompt appears if an unsaved configuration is detected.
- The reload prompt appears after the `install activate` step in this workflow. Use the `prompt-level none` keyword to automatically ignore the confirmation prompts.

Follow these steps to perform a three-step installation.

Procedure

Step 1 Use the **enable** command to enable privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 Use the **install add file** *location*: *filename* command to copy the software install package to the platform and extract the individual components of the .package file into subpackages and packages.conf files.

Example:

```
Router# install add file bootflash:<router_image>.SSA.bin
```

Note
You can copy the software install package from a remote location (through FTP, HTTP, HTTPS, or TFTP).

Step 3 (Optional) Use the **show install summary** command to display an overview of the image versions and their corresponding install state.

Example:

```
Router# show install summary
```

Step 4 Use the **install activate auto-abort-timer** *time* command to activate the previously added package and reload the platform.

Example:

```
Router# install activate auto-abort-timer 120
```

Note

- When you do a full software install, do not provide a package filename.
- In the three-step variant, the auto-abort-timer starts automatically with the `install activate` command. The default for the timer is 120 minutes. If the `install commit` command is not run before the timer expires, the install process is automatically terminated. The platform reloads and boots up with the last committed version.

Step 5 (Optional) Use the **install abort** command to terminate the software install activation and return the platform to the last committed version.

Example:

```
Router# install abort
```

Note
Use this command only when the image is in the activated state, not when the image is in the committed state.

Step 6 Use the **install commit** command to commit the new package installation and make the changes persistent across reloads.

Example:

```
Router# install commit
```

Step 7 (Optional) Use the **install rollback to committed** command to roll back the platform to the last committed state.

Example:

```
Router# install rollback to committed
```

Step 8 (Optional) Use the **install remove** {**file** *filesystem*: *filename* | **inactive**} command to delete software installation files.

Example:

```
Router# install remove inactive
```

Note

- `file` — Deletes a specific file.
- `inactive` — Deletes all the unused and inactive installation files.

Step 9 (Optional) Use the **show install summary** command to display information about the current state of the system.

Example:

```
Router# show install summary
```

Note
The output of this command varies according to the install commands run prior to this command.

a) Use the **exit** command to exit privileged EXEC mode and return to user EXEC mode.

Example:

```
Router# exit
```

The software package is successfully installed and activated on the platform. The installation is committed and persistent across reloads.

Upgrading in install mode

Use either the one-step installation or the three-step installation to upgrade the platform in install mode.

Downgrading in install mode

Downgrading in install mode is a platform management process that

- uses the **install rollback** command to revert the platform to a previous version by pointing it to the appropriate image
- requires that the target image was previously installed in install mode
- reloads the platform and boots it with the previous image.

Downgrade requirements and alternatives

The **install rollback** command has specific requirements and alternatives:

- The command succeeds only if you have not removed the previous file using the **install remove inactive** command
- Alternatively, you can downgrade by installing the older image using the **install** commands

Note The **install rollback** command succeeds only if you have not removed the previous file using the **install remove inactive** command.

Software installation termination

Software installation termination is a process that:

- stops the activation of a software package before completion,
- returns the platform to the previously running software version, and
- prevents the new software image from being committed to the system.

Termination methods

You can terminate the activation of a software package in these ways:

- When the platform reloads after activating a new image, the auto-abort-timer is triggered (in the three-step install variant). If the timer expires before issuing the **install commit** command, the installation process is terminated, and the platform reloads and boots with the last committed version of the software image.

  Alternatively, use the **install auto-abort-timer stop** command to stop this timer, without using the **install commit** command. The new image remains uncommitted in this process.

- Using the **install abort** command returns the platform to the version that was running before installing the new software. Use this command before issuing the **install commit** command.

Configuration examples

This section shows examples of using install commands.

One step installation

This example demonstrates the one-step installation process or converting from bundle mode to install mode.

```
Router#



install add file flash:c6300-universalk9.SSA.bin activate commit
install_add_activate_commit: START Mon May 30 20:45:11 UTC 2022
install_add: Adding IMG
--- Starting initial file syncing ---
Copying flash:c6300-universalk9.SSA.bin from  R0 to  R0
Info: Finished copying to the selected
Finished initial file syncing

--- Starting Add ---
Performing Add on all members
 [1] Finished Add package(s) on  R0
Checking status of Add on [R0]
Add: Passed on [R0]
Finished Add

Image added. Version: 17.09.01.0.157857

install_activate: Activating IMG
Following packages shall be activated:
/flash/c6300-mono-universalk9.SSA.pkg
```

```
/flash/c6300-rpboot.SSA.pkg

This operation may require a reload of the system. Do you want to proceed? [y/n]y


--- Starting Activate ---
Performing Activate on all members
Building configuration...
[OK] [1] Activate package(s) on  R0
 [1] Finished Activate on  R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate

--- Starting Commit ---
Performing Commit on all members
 [1] Commit package(s) on  R0
 [1] Finished Commit on  R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit operation

SUCCESS: install_add_activate_commit Mon May 30 20:48:01 UTC 2022
%PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting: reload action requested
watchdog: watchdog0: watchdog did not stop!
reboot: Restarting system


System Bootstrap, Version 3.3(REL), RELEASE SOFTWARE
Copyright (c) 1994-2021  by cisco Systems, Inc.

C6300-K9 platform with 4169728 Kbytes of main memory

MCU Version - Bootloader: 4, App: 6
MCU is in application mode.


........


Loading: bootflash:packages.conf
#

################################################################################
################################################################################
################################

%BOOT-5-OPMODE_LOG: R0/0: binos: System booted in AUTONOMOUS mode
Press RETURN to get started!

Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    17.09.01.0.157857

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

Complete three-step software installation

The three-step installation process provides a controlled method for upgrading router software by allowing verification at each stage before proceeding to the next step. Use this installation method when you need to upgrade router software with the ability to verify and roll back at each stage if needed.

Follow these steps to complete the three-step software installation.

Procedure

Step 1 Use the **install add file** *filesystem:filename* command to add the software image to the router.

Example:

```
Router# install add file flash:ir1101-universalk9.17.09.01.SPA.bin
```

Step 2 (Optional) Use the **show install summary** command to verify that the image is added and in the Inactive (I) state.

Example:

```
Router# show install summary
            [ R0 ] Installed Package(s) Information:
            State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
           --------------------------------------------------------------------------------

            Type  St   Filename/Version
           --------------------------------------------------------------------------------

            IMG   I    17.09.01.0.1
           --------------------------------------------------------------------------------

            Auto abort timer: inactive
           --------------------------------------------------------------------------------
```

Step 3 Use the **install activate** command to activate the software image. When prompted, enter **y** to proceed with the reload.

Example:

```
Router# install activate
```

Step 4 (Optional) After the system reloads, use the **show install summary** command to verify that the image is in the Activated & Uncommitted (U) state.

Example:

```
Router# show install summary
            [ R0 ] Installed Package(s) Information:
            State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
           --------------------------------------------------------------------------------

            Type  St   Filename/Version
           --------------------------------------------------------------------------------

            IMG   U    17.09.01.0.1
           --------------------------------------------------------------------------------
```

```
            Auto abort timer: inactive
           ----------------------------------------------------------------------------
```

Step 5 Use the **install commit** command to commit the software installation.

Example:

```
Router# install commit
```

Step 6 (Optional) Use the **show install summary** command to verify that the image is in the Activated & Committed (C) state.

Example:

```
Router# show install summary
            [ R0 ] Installed Package(s) Information:
            State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
           ----------------------------------------------------------------------------

            Type  St   Filename/Version
           ----------------------------------------------------------------------------

            IMG   C    17.09.01.0.1
           ----------------------------------------------------------------------------

            Auto abort timer: inactive
           ----------------------------------------------------------------------------
```

The software image is successfully installed and committed. The install summary shows the image state as "C" (Activated & Committed), confirming that the installation is complete and permanent.

Example

The following is an example of the three-step installation.

This example shows adding the image for various platforms:

```
install add file flash:ir1101-universalk9.17.09.01.SPA.bin
        install add file flash:ir1800-universalk9.17.09.01.SPA.bin
        install add file flash:ir8140-universalk9.17.09.01.SPA.bin
        install add file flash:ir8340-universalk9.17.09.01.SPA.bin
        install add file flash:c6300-universalk9.17.09.01.SPA.bin
```

This example shows the output of the add operation:

```
install_add: START Tue May 31 01:35:40 UTC 2022
        install_add: Adding IMG
        --- Starting initial file syncing ---
        Copying flash:ir1101ir1800ir8140ir8340c6300-universalk9.17.09.01.SPA.bin from R0
to R0
        Info: Finished copying to the selected
        Finished initial file syncing

        --- Starting Add ---
        Performing Add on all members
        [1] Finished Add package(s) on R0
        Checking status of Add on [R0]
        Add: Passed on [R0]
        Finished Add

        Image added. Version: 17.09.01.0.1
```

```
        SUCCESS: install_add
/flash1/ir1101ir1800ir8140ir8340c6300-universalk9.17.09.01.SPA.bin Tue May 31 01:37:10 UTC
 2022
        Router#
```

This example shows the output of the activate operation:

```
install_activate: START Tue May 31 01:37:14 UTC 2022
        install_activate: Activating IMG
        Following packages shall be activated:
        /flash/ir1101ir1800ir8140ir8340c6300-mono-universalk9_iot.17.09.01.SPA.pkg
        /flash/ir1101ir1800ir8140ir8340c6300-rpboot.17.09.01.SPA.pkg

        This operation may require a reload of the system. Do you want to proceed? [y/n]y

        --- Starting Activate ---
        Performing Activate on all members
        [1] Activate package(s) on R0
        [1] Finished Activate on R0
        Checking status of Activate on [R0]
        Activate: Passed on [R0]
        Finished Activate

        SUCCESS: install_activate Tue May 31 01:41:03 UTC 2022
        Router#
        May 31 01:41:08.684: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting:
reload action requested

        watchdog: watchdog0: watchdog did not stop!
        reboot: Restarting system

        System Bootstrap, Version 3.3(REL), RELEASE SOFTWARE
        Copyright (c) 1994-2021 by cisco Systems, Inc.

        IR1101IR1800IR8140IR8340C6300-K9 platform with 4169728 Kbytes of main memory

        MCU Version - Bootloader: 4, App: 6
        MCU is in application mode.

        ........

        Loading: bootflash:packages.conf
        #

        ##################################################################
        ##################################################################
        ########################

        Press RETURN to get started!
```

This example shows the output of the commit operation:

```
install_commit: START Tue May 31 01:47:56 UTC 2022
        --- Starting Commit ---
        Performing Commit on all members
        [1] Commit packages(s) on R0
        [1] Finished Commit packages(s) on R0
        Checking status of Commit on [R0]
        Commit: Passed on [R0]
        Finished Commit operation

        SUCCESS: install_commit Tue May 31 01:48:04 UTC 2022
```

Show the installed packages

Use these commands to display detailed information about installed packages on your router and identify which package is currently active.

Use the following command to display detailed information about an installed package:

```
Router#


show install package flash:c6300-universalk9.17.09.01.SPA.bin
  Package: c6300-universalk9.17.09.01.SPA.bin
    Size: 674114352
    Timestamp:
  Canonical path: /flash1/c6300-universalk9.17.09.01.SPA.bin

    Raw disk-file SHA1sum:
      e54ba5a59824156af7515eaf4367ebe51b920316
  Header size:     1148 bytes
  Package type:    30000
  Package flags:   0
  Header version:  3

  Internal package information:
    Name: rp_super
    BuildTime: 2022-04-27_00.47
    ReleaseDate: 2022-04-27_07.05
    BootArchitecture: arm64
    RouteProcessor: C6300
    Platform: C6300
    User: mcpre
    PackageName: universalk9
    Build: 17.09.01
    CardTypes:

  Package is bootable from media and tftp.
  Package contents:

  Package: c6300-mono-universalk9_iot.17.09.01.SPA.pkg
    Size: 673776700
    Timestamp:

    Raw disk-file SHA1sum:

    Header size:     1084 bytes
    Package type:    30000
    Package flags:   0
    Header version:  3

    Internal package information:
      Name: mono
      BuildTime: 2022-04-27_00.47
      ReleaseDate: 2022-04-27_07.05
      BootArchitecture: arm64
      RouteProcessor: C6300
      Platform: C6300
      User: mcpre
      PackageName: mono-universalk9_iot
      Build: 17.09.01
      CardTypes:

    Package is bootable from media and tftp.
```

```
    Package contents:
```

You can determine which package is active using the **show install active** command.

```
Router# show install active
[ R0 ] Active Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type St Filename/Version
--------------------------------------------------------------------------------
IMG C 17.09.01.0.1193

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

Show committed and uncommitted packages

These two show commands provide information on which packages are committed and uncommited.

```
Router# show install committed
[ R0 ] Committed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    17.09.01.0.1

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------

Router# show install uncommitted
[ R0 ] Uncommitted Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
No Uncommitted Packages
```

Remove inactive packages

This command will remove unused installation files (.conf/.pkg/.bin) from installation media.



Note This command is used to clean up the boot directory of unused installation files. This will not remove the bootable image.

```
Router#install remove inactive
install_remove: START Tue May 31 01:49:10 UTC 2022
install_remove: Removing IMG
Cleaning up unnecessary package files
No path specified, will use booted path /bootflash/packages.conf

Cleaning /flash
```

```
  Scanning boot directory for packages ... done.
  Preparing packages list to delete ...
    [R0]: /flash/packages.conf File is in use, will not delete.
   [R0]: /flash/c6300-mono-universalk9_iot.17.09.01.SPA.pkg File is in use, will not delete.

    [R0]: /flash/c6300-universalk9.17.09.01.SPA.conf File is in use, will not delete.
    [R0]: /flash/c6300-rpboot.17.09.01.SPA.pkg File is in use, will not delete.

The following files will be deleted:
    [R0]: /flash/c6300-universalk9.17.09.01.SPA.bin
    [R0]: /flash/c6300-mono-universalk9_iot.SSA.pkg
    [R0]: /flash/c6300-universalk9.SSA.conf
    [R0]: /flash/c6300-rpboot.SSA.pkg

Do you want to remove the above files? [y/n]y


Deleting file /flash/c6300-universalk9.17.09.01.SPA.bin ... done.
Deleting file /flash/c6300-mono-universalk9_iot.SSA.pkg ... done.
Deleting file /flash/c6300-universalk9.SSA.conf ... done.
Deleting file /flash/c6300-rpboot.SSA.pkg ... done.
Deleting /bootflash/.images/17.09.01.0.1.1651045630 ... done.
SUCCESS: Files deleted.

--- Starting Post_Remove_Cleanup ---
Performing REMOVE_POSTCHECK on all members
Finished Post_Remove_Cleanup
SUCCESS: install_remove Tue May 31 01:49:14 UTC 2022

Router#show install inactive
[ R0 ] Inactive Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
No Inactive Packages
```

Troubleshoot software installation using install commands

Troubleshooting the software installation and resolving other installation issues.

Software installation troubleshooting commands

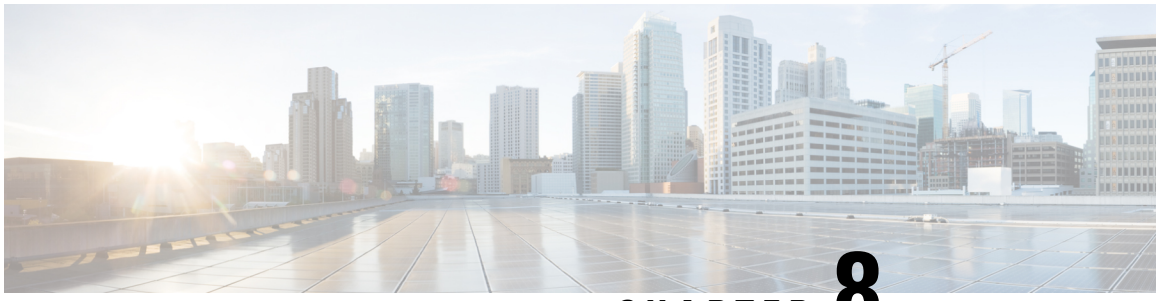
Use these show commands to view installation summary, logs, and software versions.

- **show install summary**
- **show install log**
- **show version**
- **show version running**

Other installation issue resolution

Use these commands to resolve installation issue:

- **dir** <*install directory*>
- **more location:***packages.conf*
- **show tech-support install**: this command automatically runs the **show** commands that display information specific to installation.
- **request platform software trace archive target bootflash** <*location*>: this command archives all the trace logs relevant to all the processes running on the system since the last reload, and saves this information in the specified location.



CHAPTER 8

DHCP Option 82

Table 8: Feature History Table

Feature name	Release information	Feature description
DHCP Option 82	Release 26.2.1	Starting with Cisco IOS XE Release 26.2.1, DHCP servers on routers support DHCP Option 82 relay information. The DHCP server can use the Circuit ID and Remote ID inserted by a DHCP relay agent to select an address pool for a client. This support enables predictable, location-based IP address assignment without requiring administrators to track client MAC addresses.

- DHCP Option 82, on page 105
- Prerequisites, on page 106
- Key considerations, on page 107
- Limitations, on page 107
- DHCP Option 82 packet flow, on page 107
- Hexadecimal pattern and suboption decoding, on page 108
- Configure DHCP Option 82 on server, on page 109
- Configure DHCP address pool, on page 111
- Configure DHCP relay agent, on page 112
- Configure relay chaining on relay agents, on page 113
- Verify DHCP Option 82, on page 114
- Debug DHCP Option 82 logs, on page 115

DHCP Option 82

The Dynamic Host Configuration Protocol (DHCP) Option 82 feature allows the server to allocate dynamic IP addresses based on the relay information sent by the relay agent. It enables predictive IP address assignment by the server based on network switch ports.

DHCP Option 82 enables DHCP relay agents to insert additional information such as Circuit ID, Remote ID and other configuration information into DHCP request messages before forwarding them from the client to the server.

Starting with Cisco IOS XE Release 26.2.1, DHCP servers support DHCP Option 82 on routers. This platform-independent feature includes the following suboptions:

- Circuit ID suboption: Identifies the relay agent's source interface or port where the client is connected.
- Remote ID suboption: Identifies the relay agent.

The DHCP server allows for address reservation based on Option 82 values, enabling consistent IP assignments per interface or location.

Benefits of DHCP Option 82

The use of DHCP Option 82 provides several operational and security advantages for network management such as:

- Predictable IP assignment: Enables static or predictable IP assignment for DHCP clients based on switch ports instead of MAC addresses.
- Enhanced security: Improves IP address management and security.
- Operational efficiency: Eliminates manual intervention or the need to track device MAC addresses.
- Granular control: Allows ISPs and network administrators to map specific user accounts to physical locations, enabling consistent, location-based IP assignments.

Prerequisites

Before configuring DHCP Option 82, you must ensure these requirements are met:

- A relay agent must be configured with global and VLAN-based DHCP snooping enabled.
- A DHCP server is available.
- A DHCP client is connected.
- A relay chaining topology is present wherever applicable.
- The client must be connected to the relay agent through an SVI interface for enabling VLAN-based snooping.
- Class based support must be enabled on the DHCP server.



Note You must know the hexadecimal value of each byte location in DHCP Option 82 to configure the relay-information hexadecimal command.For more information, see Circuit ID and Remote ID decode values

Key considerations

Key considerations for using DHCP Option 82 are:

- DHCP Option 82 support must be enabled on both the relay agent and DHCP server.
- You need to consider whether to use the DHCP Option 82 feature or not for predictable usage of IP assignment for DHCP clients.
- You need to consider whether to enable DHCP Option 82 globally for all interfaces or selectively on specific VLANs or interfaces only.
- You need to consider which information you want to match in the DHCP message received by the server. This can be Circuit ID, Remote ID or both.

Limitations

Limitations while using DHCP Option 82 are:

- Support must be enabled on both the relay agent and the DHCP server for it to work properly.
- If the DHCP server ignores DHCP Option 82, port-based deterministic assignment will not occur.
- It is not suitable for environments where MAC based IP address assignment is required.

DHCP Option 82 packet flow

Summary

DHCP Option 82 enables network administrators to track the location information of DHCP clients by inserting additional metadata as DHCP packets traverse relay agents. This process ensures accurate IP assignment and aids in network management.

Workflow

The DHCP packet flow through a network when DHCP Option 82 is enabled is as given:

1. The DHCP client broadcasts a DHCP request on the local network.
2. The first DHCP relay agent intercepts the broadcast and inserts the DHCP Option 82 field, which includes suboptions such as Circuit ID and Remote ID.
3. If a second relay agent resides closer to the DHCP server, the second relay agent encapsulates the DHCP Option 82 field coming from the first relay agent and forwards it to the DHCP server. This process preserves the information from both relay agents.
4. The relay agent sets the gateway IP address (giaddr) field in the DHCP packet to its own IP address. The giaddr field enables the DHCP server to identify the subnet from which the request originated.
5. The relay agent unicasts the DHCP packet to the DHCP server.

6. The DHCP server uses the DHCP Option 82 information to determine the appropriate IP address pool and configuration parameters for the client.
7. The DHCP server sends the reply to the relay agent using the giaddr address.
8. The relay agent removes the DHCP Option 82 information from the reply before forwarding the DHCP reply to the client.

Hexadecimal pattern and suboption decoding

The DHCP Option 82 hexadecimal pattern can be decoded for Circuit ID and Remote ID separately as shown in the examples.

Hexadecimal pattern sample

The Option 82 hexadecimal pattern `010600040005010a020701057065747261` consists of two primary suboptions:

- `01 06 00040005010a`: Circuit ID (Suboption 1).
- `02 07 01057065747261`: Remote ID (Suboption 2).

Field	Meaning
01	Circuit-ID suboption.
06	Circuit-ID length.
00040005010a	Circuit-ID value.
02	Remote-ID suboption.
07	Remote-ID length.
01057065747261	Remote-ID value.

Circuit ID Decode

The Circuit ID `00040005010a` identifies client's point of attachment specifying the port and VLAN of the connected client. By default, it follows the **vlan-mod-port** format.

```
Router# show cdp neighbors
         Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
         device1         Gig 1/10            175           R S I  ESR-6300- Gig 0/1/0

Router# show vlan brief
         VLAN          Name                               Status    Ports
         5    TURBINE_INTERNAL                 active    Gi1/10
```

Note Decoding the Circuit ID `00040005010a`:

- `00`: Default format (vlan-mod-port).
- `04`: Length (4 bytes).
- `0005`: VLAN ID (Hex 0005 = Decimal 5).
- `010a`: Module 1, Port 10 (Hex 0a = Decimal 10).

Remote ID Decode

The Remote ID `01057065747261` identifies the originating device (MAC address or Hostname). It can be configured to use the device hostname.

```
Router(config)# ip dhcp snooping information option format remote-id hostname
```

Note Decoding the Remote ID `01057065747261`:

- `01`: Hostname format.
- `05`: Length (5 bytes).
- `7065747261`: Hexadecimal ASCII for router.

Configure DHCP Option 82 on server

To configure DHCP Option 82 on Cisco IR router server, you need to follow these tasks:

- Enable DHCP Option 82 for DHCP address allocation on the server.
- Define the DHCP class and relay agent information patterns.
- Define the DHCP address pool on the server.
- Enable relay chaining on DHCP relay agents.
- Verify DHCP Option 82
- Debug DHCP Option 82 logs.

Follow these steps to configure DHCP Option 82 on the server.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip dhcp class** *class_name* command to enable DHCP address allocation using DHCP class.

Example:

```
Router(config)# ip dhcp class class_name
```

Note

- You can use Step 2 to enable DHCP Option 82 for DHCP address allocation on the server.
- You can disable this functionality, without deleting the DHCP class configuration.

Step 3 Use the **relay agent information** command to enter relay agent information option configuration mode and define the DHCP class and relay agent information patterns.

Example:

```
Router(dhcp-class)# relay agent information
```

Note
The `relay agent information` command validates only the presence of Option 82 in the DHCP packet. It does not evaluate the Circuit ID or Remote ID. As a result, any DHCP packet containing Option 82, regardless of its source port or relay agent will match this DHCP class and be assigned an IP address.

Step 4 (Optional) Use the **relay-information hex** *pattern* [*] [**mask** *mask*] command to optionally specify a hexadecimal value for the full relay information option.

Example:

```
Router(dhcp-class-relayinfo)# relay-information hex 010600040005010a020701057065747261
```

Example:

```
Router(dhcp-class-relayinfo)# relay-information hex 01030a0b0c02050000000123 mask
00000000000000000ffffffffffffffff
```

Example:

```
Router(dhcp-class-relayinfo)# relay-information hex 010600040005010a*
```

Note

- The Option 82 hexadecimal pattern includes two sub options: one for Circuit ID and the other for Remote ID.
- Each suboption can be decoded to represent the suboption type, length, VLAN, module port, client interface and MAC address.
- You can match the hexadecimal pattern either fully or partially for specific scenarios of IP assignment.
- You can use the mask option to mask either the Circuit ID or Remote ID partially.
- You can use `*` as a wildcard pattern at the end of the hexadecimal pattern to match only the beginning of the relay information value.
- If you omit this step, no pattern is configured and it is considered a match to any relay agent information option value, but the relay information option must be present in the DHCP packet.

- Repeat Step 4 for each DHCP class you need to configure.

What to do next

For more information, see Configuring DHCP Address Allocation Using Option 82.

Configure DHCP address pool

Follow these steps to configure and define DHCP address pool for DHCP address allocation.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip dhcp pool** *name* command to configure a DHCP address pool on a DHCP server and enter DHCP pool configuration mode.

Example:

```
Router(config)# ip dhcp pool ABC
```

Step 3 Use the **network** *network-number* [*mask* |*prefix-length*] command to configure the subnet number and mask for a DHCP address pool on the DHCP server.

Example:

```
Router(dhcp-pool)# network 10.0.20.0 255.255.0.0
```

Step 4 Use the **class** *name* command to associate a class with a pool and enters DHCP pool class configuration mode.

Example:

```
Router(dhcp-pool)# class CLASS1
```

Note
This command will also create a DHCP class if the DHCP class is not yet defined.

Step 5 Use the **address range** *start-ip end-ip* command optionally to set an address range for a DHCP class in a DHCP server address pool.

Example:

```
Router(dhcp-pool-class)# address range 10.0.20.1 10.0.20.100
```

Note

- Repeat Step 5 for each DHCP class you need to associate to the DHCP pool.

- If this command is not configured for a class, the default value is the entire subnet of the pool.

Configure DHCP relay agent

Use this task to configure the client-facing interface as a relay agent port on the router.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip dhcp snooping** command to enable the DHCP snooping feature globally

Example:

```
Router(config)# ip dhcp snooping
```

Step 3 Use the **ip dhcp snooping vlan** *vlan-id* command to enable DHCP snooping for VLAN.

Example:

```
Router(config)# ip dhcp snooping vlan 5
```

Step 4 Use these steps to configure SVI:

a) Use the **interface** *vlan-id* command to select the SVI corresponding to the client VLAN.

Example:

```
Router(config)# interface Vlan5
```

b) Use the **ip address** *ip-address subnet-mask* command to assign an IP address and subnet mask to the SVI.

Example:

```
Router(config-if)# ip address 10.166.1.1 255.255.0.0
```

c) Use the **ip helper-address** *ip-address* command to specify the IP address of the DHCP server.

Example:

```
Router(config-if)# ip helper-address 110.1.1.1
```

Step 5 Use the **interface** command to select the specific interface connected to the client.

Example:

```
Router(config)# interface GigabitEthernet1/10
```

Step 6 Use the **description** *client* command to provide a label for the client connection.

Example:

```
Router(config-if)# description CLIENT-1
```

Step 7 Use the **switchport mode access** command to set the interface to access mode.

Example:

```
Router(config-if)# switchport mode access
```

Step 8 Use the **switchport access vlan** command to assign the interface to the client VLAN.

Example:

```
Router(config-if)# switchport access vlan 5
```

Step 9 (Optional) Use these commands to define custom Circuit ID and Remote ID values, overriding the defaults:

a) (Optional) Use the **ip dhcp snooping vlan** *vlan-id* **information option format-type circuit-id string** *circuit-id-string* command to override the default Circuit ID (**vlan mod port**) with a user-defined string.

Example:

```
Router(config-if)# ip dhcp snooping vlan 5 information option format-type circuit-id
string abc
```

b) (Optional) Use the **ip dhcp snooping information option format remote-id string** *remote-id-string* command to override the default Remote ID (MAC address) with a user-defined string.

Example:

```
Router(config)# ip dhcp snooping information option format remote-id string abc
```

c) Use the **exit** command to return to global configuration mode.

Example:

```
Router(config-if)# exit
```

What to do next

For more information, see Configuring the Cisco IOS XE DHCP Relay Agent.

Configure relay chaining on relay agents

Use this task to configure relay chaining when multiple relay agents are used in the network. Enabling encapsulation on the secondary relay agent ensures that the original DHCP Option 82 data is preserved as the request is forwarded to the DHCP server.

Before you begin

Note

- Ensure DHCP snooping is enabled on both relay agents for their respective client VLANs.
- Relay chaining requires that DHCP snooping be enabled on both relay agents for their respective client VLANs. Ensure that interfaces are correctly configured between the relay agents to allow the passage of DHCP traffic.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip dhcp relay information option vpn** command to enable the inclusion of VPN information in the relay agent inforamtion.

Example:

```
Router(config)# ip dhcp relay information option vpn
```

Step 3 Use the **ip dhcp relay information policy encapsulate** command to enable the encapsulation policy on the secondary relay agent.

Example:

```
Router(config)# ip dhcp relay information policy encapsulate
```

Note
In a double relay agent scenario, this configuration must be applied on the relay agent closest to the server.

Step 4 Use the **exit** command to return to privileged EXEC mode.

Example:

```
Router(config)# exit
```

An example of an encapsulated hexadecimal pattern format received by server in relay chaining is as given:

```
091f000000091a0a18bea80a0152120106000401f401050208000668d972ad3540020c020a00003c3c3c010e000000
```

Verify DHCP Option 82

Use these steps to verify DHCP Option 82 configuration.

Procedure

Step 1 Use these commands on the DHCP server side:

a) Use the **more system:running-config** command to display the running configuration.

Example:

```
Router# show running-config
```

Note
This command will display the configured DHCP classes and DHCP pools.

b) Use the **show ip dhcp binding** command to display the client's MAC IP address.

Example:

```
Router# show ip dhcp binding
```

Step 2 Use these commands on the DHCP relay agent side:

a) Use the **show ip dhcp snooping binding** command to verify the relay agent's snooping bindings.

Example:

```
Router# show ip dhcp snooping binding
```

b) Use the **show ip dhcp snooping** command to validate the Circuit ID and Remote ID.

Example:

```
Router# show ip dhcp snooping
```

Debug DHCP Option 82 logs

Use this task to troubleshoot DHCP Option 82 configurations and derive the required hexadecimal patterns for class matching by enabling debug logs on both the DHCP server and the relay agent.

Procedure

Step 1 Use the **debug ip dhcp server** command on the DHCP server to monitor server-side events, classes, and packets.

Example:

```
Router# debug ip dhcp server event
         Router# debug ip dhcp server class
         Router# debug ip dhcp server packet
```

Step 2 Use the **debug ip dhcp snooping** command on the relay agent to monitor snooping events, agent processing, and packets.

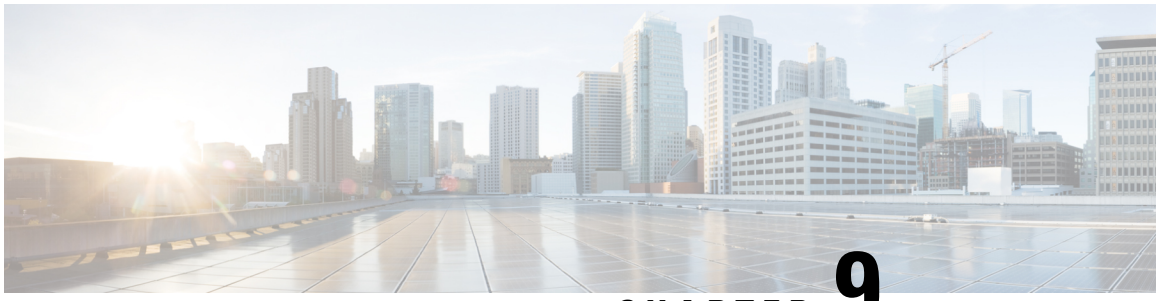
Example:

```
Router# debug ip dhcp snooping agent
         Router# debug ip dhcp snooping event
         Router# debug ip dhcp snooping packet
```

Note
You can analyze the generated logs to identify the `relay-information` hexadecimal pattern being received by the server.

Example:

```
*May 13 2026 08:25:19.629: DHCPD: Searching for a match to 'relay-information
010600040005010a020701057065747261' in class class_turbine
          *May 13 2026 08:25:19.629: DHCPD: input pattern 'relay-information
010600040005010a020701057065747261' matches class CLASS1
```



CHAPTER 9

IKEv2 and IPsec PFS Post Quantum Cryptography

Table 9: Feature History Table

Feature name	Release information	Feature description
IKEv2 and IPsec PFS Post Quantum Cryptography	Release 26.2.1	Cisco IoT routers can combine a traditional Diffie-Hellman key exchange with the NIST-standardized Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) to establish keys for IKEv2/IPsec sessions. This hybrid approach helps protect recorded VPN traffic against future harvest-now, decrypt-later attacks.

- IKEv2 and IPsec PFS Post Quantum Cryptography, on page 117
- Benefits of PQC, on page 118
- Prerequisites, on page 118
- Limitations, on page 118
- ML-KEM algorithm and parameter sets, on page 119
- PFS negotiation behavior, on page 120
- Configure an IKEv2 proposal for PQC, on page 120
- Configure Perfect Forward Secrecy with PQC, on page 122
- Verify the PQC hybrid key exchange, on page 123
- Troubleshooting PQC hybrid key exchange, on page 124

IKEv2 and IPsec PFS Post Quantum Cryptography

Quantum-Safe Encryption refers to cryptographic techniques designed to protect network communications against attacks by computers and future cryptographically relevant quantum computers.

Post-Quantum Cryptography (PQC) is part of this broader security approach. PQC includes quantum-resistant algorithms for functions such as key establishment and digital signatures. These algorithms help mitigate harvest-now, decrypt-later attacks, in which an attacker records encrypted traffic and attempts to decrypt it later using a sufficiently capable quantum computer.

PQC does not replace existing public-key algorithms. Instead, it works alongside them to create a quantum-safe hybrid key exchange.

The Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) is a National Institute of Standards and Technology (NIST)-standardized post-quantum key-encapsulation mechanism. ML-KEM enables two peers to establish a shared secret that can be used to derive cryptographic session keys.

On supported Cisco routers, ML-KEM is combined with a traditional key-exchange algorithm to provide a hybrid key establishment for IKEv2/IPsec Perfect Forward Secrecy (PFS). The hybrid approach protects the negotiated session as long as at least one of the constituent key-establishment mechanisms remains secure.

For more information on PQC on Cisco routers with IKEv2 sessions, see Security and VPN Configuration Guide.

For more information on Quantum-Safe Encryption and the related protocols, see: https://www.cisco.com/c/en/us/td/docs/routers/ios/config/17-x/sec-vpn/b-security-vpn/m-sec-cfg-quantum-encryption-ppk.html.



Note Postquantum preshared keys (PPKs), including dynamic PPKs obtained through the Secure Key Integration Protocol (SKIP), can be used with ML-KEM-based hybrid key establishment.

Benefits of PQC

You can use the PQC feature to strengthen the security of your IPsec and IKEv2 connections with quantum-safe, future-proof protection for traffic between Cisco IoT routers.

The PQC hybrid key exchange provides these benefits:

- Future-proof security: Protects traffic against future quantum-computer-based attacks.
- Defense in depth: Combines a classical DH algorithm with ML-KEM, so the session stays secure even if one algorithm is broken.
- Smooth migration: Supports phased migration in existing networks through the optional fallback keyword.

Prerequisites

Before you configure the PQC hybrid key exchange, ensure that the following requirements are met:

- Both peers support PQC, or the optional keyword is used to allow fallback during migration.
- IKEv2/IPsec is configured and operational between the peers.

Limitations

The limitations with respect to using PQC are:

- IKEv2 always uses hybrid key exchange: a combination of Diffie-Hellman (DH) and ML-KEM algorithms. Pure-PQC key exchange is not supported for IKEv2.

- The encapsulation keys and ciphertexts used by ML-KEM-768 and ML-KEM-1024 may cause IKEv2 messages to exceed typical path maximum transmission units (MTUs), resulting in IP fragmentation and potential packet drops. To avoid IP fragmentation, enable IKEv2 fragmentation and set the IKEv2 fragmentation MTU to 1400 bytes.
- This feature provides post-quantum key establishment only. It does not provide post-quantum peer authentication. IKEv2 peers continue to use the configured authentication method, such as a preshared key or certificate-based authentication.
- ML-KEM establishes shared keying material; it does not encrypt user traffic. The encryption algorithm configured in the IPsec transform set, such as AES-CBC or AES-GCM, continues to encrypt the user traffic.

Note License, platform, and software support: This feature does not require an additional feature license. ML-KEM-based quantum-safe key establishment is supported on all Cisco IoT routers with Cisco IOS XE Release 26.2.1.

Supported routers

The supported routers are:

- Cisco Catalyst IR1101 Rugged Series Router,
- Cisco Catalyst IR1800 Rugged Series Router,
- Cisco Catalyst IR8340 Rugged Series Router,
- Cisco Catalyst IR8140 Heavy Duty Router, and
- Cisco ESR6300 Embedded Series Router.

Phased migration in Hub-and-Spoke deployments

If an older responder (hub) receives a PQC proposal from upgraded spokes (initiators), the negotiation fails.

To avoid this, it is recommended that you always upgrade the hub first and configure **optional** in the hub's IKEv2 proposal.

When any issue occurs, the responder logs the following errors:

IKEv2-INTERNAL: Transform type 6 invalid.

IKEv2-ERROR: Failed to parse the packet: The peer's proposal is invalid.

ML-KEM algorithm and parameter sets

Cisco IoT routers support the three National Institute of Standards and Technology (NIST)-standardized ML-KEM parameter sets: ML-KEM-512, ML-KEM-768, and ML-KEM-1024. During the key exchange, the peers exchange an encapsulation key (public key) and a ciphertext.

Table 10: ML-KEM parameter sets (sizes in bytes)

Parameter	ML-KEM-512	ML-KEM-768	ML-KEM-1024
Encapsulation key size (public key)	800	1184	1568
Decapsulation key size (private key)	1632	2400	3168
Shared secret key size	32	32	32
Ciphertext size	768	1088	1568

PFS negotiation behavior

Table 11: PFS negotiation behavior for Cisco IOS XE to Cisco IOS XE devices

Initiator/ Responder	no set pfs	set pfs	set pfs <group>	set pfs <group> pqc <mlkem>
set pfs	UP (ML-KEM inherit)	UP (ML-KEM inherit)	UP (DH+ML-KEM inherit)	UP (DH+ML-KEM inherit)
set pfs <*group*>	UP (DH)	DOWN	UP (DH)	DOWN
set pfs <*group*> **pqc** <*mlkem*>	UP (DH+ML-KEM)	UP (DH+ML-KEM inherit)	UP (DH+ML-KEM)	UP (DH+ML-KEM)

Note During initial session establishment, `CHILD_SA` output might display `PFS: N`. After the `CHILD_SA` rekeys, verify the negotiated ML-KEM and DH PFS settings.

Configure an IKEv2 proposal for PQC

Follow these steps to enable ML-KEM algorithm in an IKEv2 proposal.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **crypto ikev2 fragmentation mtu** *mtu-size* command to enable IKEv2 fragmentation and set the IKEv2 fragmentation MTU to 1400 bytes.

Example:

```
Router(config)# crypto ikev2 fragmentation mtu 1400
```

Step 3 Use the **crypto ikev2 proposal** *name* command to create or edit an IKEv2 proposal.

Example:

```
Router(config)# crypto ikev2 proposal prop1
```

Step 4 Use the **encryption** *aes-cbc-key* command to configure AES key in CBC mode.

Example:

```
Router(config-ikev2-proposal)# encryption aes-cbc-256
```

a) Use the **integrity** *integrity-algorithm* command to configure the IKEv2 integrity algorithm.

Example:

```
Router(config-ikev2-proposal)# integrity sha512
```

b) Use the **group** *group-number* command to configure Diffie-Hellman group for classical key exchange used with ML-KEM.

Example:

```
Router(config-ikev2-proposal)# group 19
```

Step 5 Use the **pqc mlkem1024 mlkem768 mlkem512 optional** command to enable one or more ML-KEM algorithm parameter sets.

Example:

```
Router(config-ikev2-proposal)# pqc mlkem1024 mlkem768 mlkem512 optional
```

Note
The proposal enables **pqc mlkem768 optional** by default. The **optional** keyword allows the SA to fall back to a classical (non-PQC) negotiation if the peer does not support PQC, which supports coexistence during phased migration.

Step 6 Use these commands to create the IKEv2 policy and attach the proposal:

a) Use the **crypto ikev2 policy** *policy* command to create the IKEv2 policy.

Example:

```
Router(config)# crypto ikev2 policy pol1
```

b) Use the **match fvrf any** command to apply the IKEv2 policy to negotiations through any front-door VRF (FVRF).

Example:

```
Router(config-ikev2-policy)# match fvrf any
```

c) Use the **proposal** *proposal-name* command to add the PQC-enabled proposal to the IKEv2 policy.

Example:

```
Router(config-ikev2-policy)# proposal prop1
```

The **proposal** *proposal* command makes the PQC-enabled proposal available during IKEv2 negotiation for sessions that match this policy.

Step 7 Use these commands to associate the existing IKEv2 profile with the IPsec profile:

a) Use the **crypto ipsec profile** *profile-name* command to create an IPsec profile.

Example:

```
Router(config)# crypto ipsec profile gen
```

b) Use the **set ikev2-profile** *ikev2-profile-name* command to associate an existing IKEv2 profile with the IPsec profile.

Example:

```
Router(ipsec-profile)# set ikev2-profile prof1
```

Note
The profile and IPsec transform set can already exist, if not you can create it using the create IPsec profile option given in step 7.

Step 8 Use these commands to apply the IPsec profile to the tunnel:

a) Use the **interface tunnel** *tunnel-interface-number* command to create a tunnel interface.

Example:

```
Router(config)# interface tunnel 1
```

b) Use the **tunnel protection ipsec profile** *ipsec-profile-name* command to protect the tunnel with IPsec profile.

Example:

```
Router(config-if)# tunnel protection ipsec profile gen
```

Step 9 Use the **end** command to return to privileged EXEC mode.

Example:

```
Router(config-if)# end
```

Configure Perfect Forward Secrecy with PQC

Perfect Forward Secrecy (PFS) adds extra randomness to derive session keys during each key exchange. With PQC, the key exchange can use DH, ML-KEM, or DH + ML-KEM. PFS is optional and can be disabled.

Follow these steps to configure PFS with PQC.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **crypto ipsec profile** *name* command to enter the IPsec profile.

Example:

```
Router(config)# crypto ipsec profile P1
```

Step 3 Use the **set pfs** *group* **pqc** *mlkem-algorithm* command to enable PFS, specify a DH group, and an ML-KEM algorithm, or both.

Example:

```
Router(ipsec-profile)# set pfs group19 pqc mlkem1024
```

Note
When **set pfs** is used:

- Without PQC: Starting from Cisco IOS XE Release 26.2.1 onwards, the DH group is inherited from the IKEv2 SA.
- With PQC: The DH group is inherited from the IKEv2 SA. Exchanges are optimized for Cisco IOS XE to Cisco IOS XE deployments using an ML-KEM-only IPsec Child Security Association exchange.

Verify the PQC hybrid key exchange

Follow these steps to verify that the PQC hybrid key exchange is active.

Procedure

Step 1 Use the **show crypto ikev2 sa detail** command to verify the IKEv2 SA and the PQC key exchange algorithm.

Example:

```
Router# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

Tunnel-id Local              Remote             fvrf/ivrf   Status
1         10.0.149.203/500   10.0.149.217/500   none/none   READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:19, Auth sign: PSK, Auth
verify: PSK
PQC Key Exchange: ML-KEM-1024
...
Quantum-safe Encryption using PQC: ML-KEM-1024
PEER TYPE: IOS-XE
```

Step 2 Use the **show crypto ipsec sa** command to verify the IPsec SA. The PQC key exchange appears after the first IPsec SA rekey.

Example:

```
Router# show crypto ipsec sa
...
PFS (Y/N): Y, DH group: group19, PQC Key Exchange: ML-KEM-1024
```

Note
The PQC key exchange value in the IPsec SA output appears only after an IPsec SA rekey.

When the show commands display the configured ML-KEM algorithm and the tunnel status is READY, the quantum-safe hybrid key exchange is active.

Troubleshooting PQC hybrid key exchange

Use these debug commands to troubleshoot PQC hybrid key exchange issues on a Cisco IoT router.

Follow these steps to debug the PQC hybrid key exchange.

Procedure

Step 1 Use the **debug crypto ikev2** commands to debug IKEv2 negotiation, errors, packets, and internal events.

Example:

```
Router# debug crypto ikev2
Router# debug crypto ikev2 error
Router# debug crypto ikev2 packet
Router# debug crypto ikev2 internal
```

Step 2 Use the **debug crypto kmi** command to debug Key Management Infrastructure (KMI) events.

Example:

```
Router# debug crypto kmi
```

Step 3 Use the **debug crypto socket** command to debug crypto socket events.

Example:

```
Router# debug crypto socket
```

Step 4 Use the **debug crypto ipsec** commands to debug IPsec negotiation, errors, messages, and state changes.

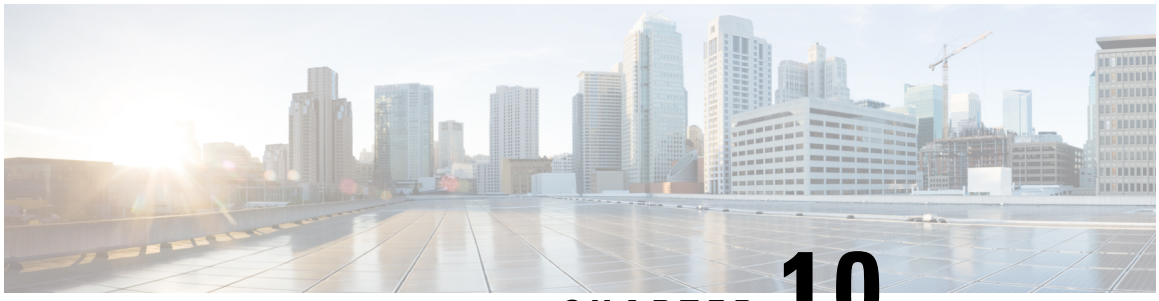
Example:

```
Router# debug crypto ipsec
Router# debug crypto ipsec error
Router# debug crypto ipsec message
Router# debug crypto ipsec state
```

Step 5 Use the **debug tunnel protection** command to debug tunnel protection events.

Example:

```
Router# debug tunnel protection
```



CHAPTER 10

Segment Routing over MPLS

Table 12: Feature History Table

Feature name	Release information	Feature description
Segment Routing over MPLS	Release 26.2.1	Starting from Cisco IOS XE Release 26.2.1, you can configure Segment Routing over Multiprotocol Label Switching (SR-MPLS) on supported WAN ports. SR-MPLS encodes an ordered list of segments in the packet and integrates with MPLS services such as Layer 3 VPN and Ethernet VPN.

- Segment Routing over MPLS, on page 125
- Prerequisites, on page 126
- Configure Segment Routing over MPLS, on page 126
- Configure Segment Routing MPLS on OSPF network, on page 127
- Configure Prefix-SID for OSPF, on page 128
- Verify SR-MPLS configuration, on page 129

Segment Routing over MPLS

Segment Routing over Multi-protocol Label Switching (SR-MPLS) on WAN ports is a routing architecture that provides a flexible and scalable approach to source routing. It allows the source to define a path and encode it in the packet header as an ordered list of segments. Each segment is identified by a Segment ID (SID). When operating using the MPLS dataplane, segment routing integrates with the multi-service capabilities of MPLS, including Layer 3 VPN (L3VPN) and Ethernet VPN (EVPN).

Note SR-MPLS is applied specifically to the WAN ports of the supported routers. Ensure the WAN interfaces are active and configured with the appropriate routing protocols (OSPF or IS-IS) before proceeding.

Benefits of using SR-MPLS

SR-MPLS helps in:

- Scalability: The network does not need to maintain per-application or per-flow states.
- Traffic Engineering: Offers lower latency and more effective bandwidth utilization than traditional MPLS networks.
- Automatic Protection: Provides automatic traffic protection against link and node failures without requiring additional signaling.

Prerequisites

Before configuring SR-MPLS on your router, you must ensure that:

- Cisco Catalyst IR1101 or IR8340 routers with active WAN ports are used.
- The software image supports SR-MPLS configuration for these routers.
- Intermediate System to Intermediate System (IS-IS) and Open Shortest Path First (OSPF) routing protocols are enabled and configured for SR-MPLS operation.

Configure Segment Routing over MPLS

Use this task to enable SR-MPLS mode.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **segment-routing mpls** command to enable segment routing MPLS.

Example:

```
Router(config)# segment-routing mpls
```

Step 3 Use the **global-block** *start-range end-range* command to enable Segment Routing Global Block (SRGB) to define the label range for Prefix-SIDs.

Example:

```
Router(config)# global-block 16000 23999
```

For more information, see Segment Routing Global Block

Step 4 Use the **connected-prefix-sid-map** command to enter the sub-mode for configuring address-family specific mappings.

Example:

```
Router(config-srmpls)# connected-prefix-sid-map
```

Step 5 Use the **address-family** *ipv4* command to specify the IPv4 address family.

Example:

```
Router(config-srmpls-conn)# address-family ipv4
```

Step 6 Use the *ip-address/mask* **index** *sid-index* **range** *range-value* command to associate a Segment Identifier (SID) with the IP address.

Example:

```
Router(config-srmpls-conn-af)# 4.4.4.4/32 index 4 range 1
```

Step 7 Use the **exit-address-family** command to exit the address family configuration mode and return to the **connected-prefix-sid-map** sub-mode.

Example:

```
Router(config-srmpls-conn-af)# exit-address-family
```

Configure Segment Routing MPLS on OSPF network

Use this task to define the network topology, select OSPF as the control protocol, and configure the necessary SID mappings.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **router ospf** *process-id* command to enable the OSPF routing process.

Example:

```
Router(config)# router ospf 10
```

Step 3 Use the **router-id** *id* command to manually configure the OSPF router ID.

Example:

```
Router(config-router)# router-id 10.0.0.1
```

Step 4 Use the **segment-routing mpls** command to enable segment routing under the OSPF process.

Example:

```
Router(config-router)# segment-routing mpls
```

Step 5 Use the **segment-routing area** *area-id* **mpls** command to configure segment routing MPLS mode for a specific OSPF area.

Example:

```
Router(config-router)# segment-routing area 0 mpls
```

Configure Prefix-SID for OSPF

This task allows you to map a specific Prefix-SID to a loopback or interface address. Mapping the Prefix-SID to a loopback address creates a Node-SID, which serves as the node's unique identity. This enables other routers to forward traffic to this specific prefix using the associated SID label.

Before you begin

Ensure that segment routing is enabled on the corresponding address family.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **segment-routing mpls** command to enable segment routing MPLS mode.

Example:

```
Router(config)# segment-routing mpls
```

Step 3 Use the **connected-prefix-sid-map** command to enter the sub-mode for configuring address-family specific mappings.

Example:

```
Router(config-srmpls)# connected-prefix-sid-map
```

Step 4 Use the **address-family** *ipv4* command to specify the IPv4 address family.

Example:

```
Router(config-srmpls-conn)# address-family ipv4
```

Step 5 Use the *ip-address/mask* **index** *sid-index* **range** *range-value* command to associate a SID with the specified IPv4 prefix.

Example:

```
Router(config-srmpls-conn-af)# 10.1.1.1/32 index 100 range 1
```

Step 6 Use the **exit** command to exit to global configuration mode.

Example:

```
Router(config-srmpls-conn-af)# exit
```

Verify SR-MPLS configuration

Use this task to validate connectivity, path establishment, and resilience of your SR-MPLS deployment.

Before you begin

Verify the node SID reachability to ensure each node in the SR-MPLS domain can reach its designated node SID.

Procedure

Step 1 Use the **show segment-routing mpls connected-prefix-sid-map local ipv4** command to verify the SR-MPLS configuration on the router.

Example:

```
Router# show segment-routing mpls connected-prefix-sid-map local ipv4
```

Step 2 Use the **show mpls forwarding-table** command to check the MPLS label table.

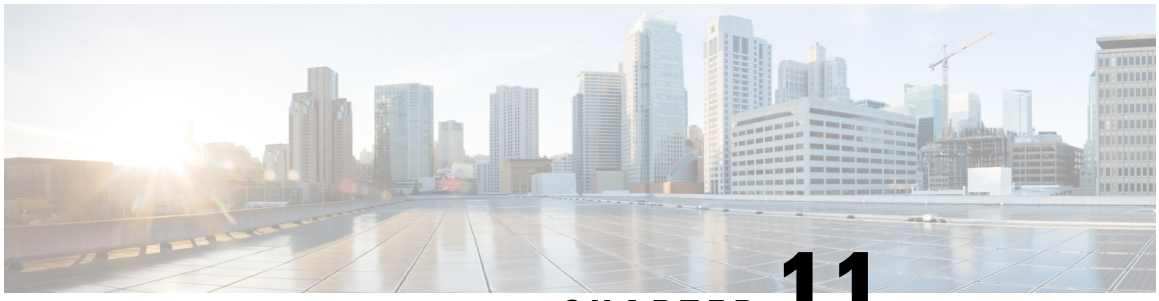
Example:

```
Router# show mpls forwarding-table
```

Step 3 Use the **ping mpls** command to validate the end-to-end connectivity between critical devices.

Note

- You can simulate link or node failures to ensure traffic is automatically rerouted along available alternate paths.
- You can also test adjacency SID reachability between adjacent nodes.



CHAPTER 11

Post Quantum Cryptography for SSHv2 sessions

Table 13: Feature History Table

Feature name	Release information	Feature description
Post Quantum Cryptography for SSHv2 sessions	Release 26.2.1	Cisco IoT routers can combine the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) with classical elliptic-curve key-exchange algorithms to establish keys for SSHv2 sessions.

- Implementation of Post Quantum Cryptography, on page 131
- Overview, on page 132
- Factors to consider for deploying PQC, on page 132
- Protocol requirements, on page 132
- Restrictions for PQC, on page 132
- PQC negotiation scenarios , on page 133
- Configure PQC for SSHv2 sessions, on page 133
- Verify PQC for SSHv2 sessions, on page 134
- Key elements of PQC, on page 134
- Supported platforms, on page 136

Implementation of Post Quantum Cryptography

Cisco utilizes a hybrid approach to implement Post Quantum Cryptography (PQC). The hybrid approach to PQC combines a quantum-resistant algorithm, such as the module-lattice based ML-KEM (e.g., mlkem768), with a classical elliptic curve key exchange algorithm like NIST P-256 or X25519, along with a hash function such as SHA-256.

This combination provides enhanced security by protecting against both classical and quantum attacks while maintaining compatibility and performance with existing cryptographic protocols.

Cisco implements this hybrid approach in protocols like SSH allowing configurations that enforce the use of ML-KEM algorithms for quantum resistance without falling back to classical-only key exchanges.

Overview

PQC refers to cryptographic algorithms designed to be secure against attacks from quantum computers which can break many classical encryption methods. Current widely used encryption methods—such as Diffie-Hellman (DH), and Elliptic Curve Diffie-Hellman (ECDH) — are vulnerable to being broken as quantum computing technology advances.

Implementing quantum-safe encryption now is critical for the long-term confidentiality and integrity of sensitive communications, particularly for government, military, and financial sectors.

Factors to consider for deploying PQC

Review these factors before deploying PQC.

Backward compatibility with classical algorithms
If a device is only configured for PQC and the SSH client does not support PQC, the SSH session will fail and no connection will be established.
Impact on computational time
The key size in PQC implementations may have an impact on the computational time due to the mathematical complexity and processing requirements of quantum-resistant algorithms. Larger key sizes generally increase the computational workload. PQC implementations must consider hardware acceleration and software optimizations to mitigate the increased computational time due to larger key sizes.

Protocol requirements

Protocol requirements

SSHv2 :

To establish secure, encrypted, and remote command-line access to servers and network devices over unsecured network use SSH version 2.0. PQC cannot be implemented on SSHv1 due to severe security vulnerabilities and it does not support the modern, complex key exchange mechanisms required to resist quantum computer attacks.

Key Pairs :

Ensure that RSA key pair or EC key pair for the SSH server is generated using crypto key generate command.

Restrictions for PQC

Implementing PQC is not supported on devices that operate in Federal Information Processing Standards (FIPS) mode.

PQC negotiation scenarios

During the SSH protocol connection setup, the client and server negotiate which cryptographic algorithms to use for securing the session. This negotiation process is a basic example of how algorithm negotiation works in SSH and is applicable to any similar negotiation process.

The algorithms are considered in the order they are configured on the device, meaning the priority or preference of algorithms is determined by the sequence in which they are listed in the configuration. The negotiation proceeds by trying the algorithms in that configured order until a mutually supported algorithm is found and agreed upon for use in the session. The server must support at least one algorithm from the client's list for successful negotiation.

Table 14: sfds

Client configuration	Server configuration	Negotiated algorithm
Classical, Hybrid	Classical, Hybrid	Classical
Hybrid, Classical	Hybrid, Classical	Hybrid
Classical, Hybrid	Hybrid	Hybrid
Hybrid, Classical	Classical	Classical
Classical	Classical, Hybrid	Classical
Hybrid	Classical, Hybrid	Hybrid
Classical	Hybrid	None
Hybrid	Classical	None
Mixed (No overlap)	Mixed (No overlap)	None

Configure PQC for SSHv2 sessions

Use these commands to set up devices as an SSH server and an SSH client. This allows the device to accept connection requests from remote SSH clients, enabling secure remote management of the device through SSH connections.

Note The PQC algorithms are not enabled by default and must be explicitly enabled.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ip ssh server algorithm kex** *algorithm* command to configure the PQC based key-exchange algorithms that the SSH server supports.

Example:

```
Router(config)# ip ssh server algorithm kex mlkem768nistp256-sha256
```

Step 3 Use the **ip ssh client algorithm kex** *algorithm* command to configure the PQC based key-exchange algorithms that SSH client supports.

Example:

```
Router(config)# ip ssh client algorithm kex mlkem768x25519-sha256
```

This command is used to define the order and selection of key exchange (KEX) algorithms that the SSH client will use during the SSH handshake process.

Verify PQC for SSHv2 sessions

Use this section to understand how to verify PQC implementation for SSHv2.

Command	Purpose	What to look for in output
show ip ssh	Displays the status of the SSH server, its configuration details.	In the output of this command, look for these details that indicate PQC implementation is successful: KEX algorithms: **mlkem1024nistp384-sha384,** **mlkem768nistp256-sha256,** **mlkem768x25519-sha256**

This example shows the response when a PQC algorithm command is entered on a platform that does not support PQC.

```
Router(config)# ip ssh server algorithm kex mlkem1024nistp384-sha384 ecdh-sha2-nistp256
Router(config)#
000205: Jun 18 16:41:43.086: %SSH-3-KEX_PQC_NOT_SUPP: PQ/T hybrid key exchange algorithms
are not supported on this platform.
Router(config)#
```

Key elements of PQC

PQC is implemented by using the algorithms available as part of Module-lattice based Cryptography. Module-lattice based Cryptography is a type of quantum-resistant public-key cryptography that uses structured, high-dimensional grid points (lattices) to secure data. Cisco implements PQC into protocols like SSH using ML-KEM.

These algorithms are resistant to quantum attacks and are used for secure key establishment, encryption, and digital authentication or signatures.

This table lists the areas in which each algorithm can be used:

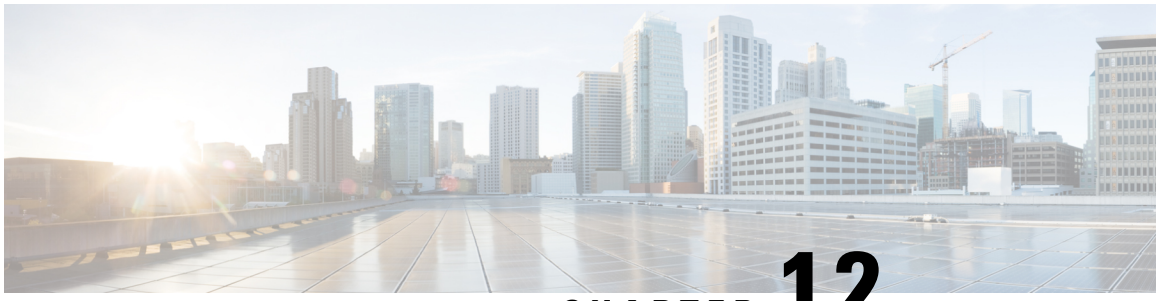
ML-KEM algorithm	What this algorithm means	Security Level
mlkem768x25519-sha256	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. X25519 is a widely used classical elliptic curve Diffie-Hellman key exchange algorithm. SHA-256 is the hash function used for cryptographic hashing in the key exchange process. In summary, mlkem768x25519-sha256 is a hybrid PQC key exchange algorithm combining the ML-KEM 768-bit lattice-based scheme with classical X25519 and SHA-256, used to secure key exchanges against quantum and classical threats in Cisco's PQC-enabled network protocols.	The security level of this implementation is balanced and is best suited for deployments where performance metrics are critical.
mlkem768nistp256-sha256	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. NIST P-256 is a widely used classical elliptic curve Diffie-Hellman key exchange algorithm standardized by NIST. SHA-256 is the hash function used for cryptographic hashing in the key exchange process. In summary, mlkem768x25519-sha256 is a hybrid PQC key exchange algorithm combining the ML-KEM 768-bit lattice-based scheme with classical X25519 and SHA-256, used to secure key exchanges against quantum and classical threats in Cisco's PQC-enabled network protocols.	The security level of this implementation is good and is best suited for most deployments.

ML-KEM algorithm	What this algorithm means	Security Level
mlkem1024nistp384-sha384	ML-KEM 768 is a PQC algorithm based on module-lattice cryptography, providing resistance against quantum attacks. 1024 is the highest security parameter set, equivalent to the security level of AES-256. NIST P-384 is the traditional (classical) component. It uses Elliptic Curve Diffie-Hellman (ECDH) with the NIST P-384 curve to maintain security against current non-quantum threats. In summary, mlkem1024nistp384-sha384 represents a hybrid cryptographic proposal combining a quantum-resistant ML-KEM algorithm with classical elliptic curve cryptography (NIST P-384) and SHA-384 hashing. This hybrid approach ensures security against both classical and quantum attacks by negotiating keys using both classical and post-quantum algorithms.	The security level of this implementation is the highest and is best suited for deployments where maximum security is needed.

Supported platforms

PQC can be configured only on these platforms in autonomous mode:

- Cisco Catalyst IR1101 Rugged Series Routers,
- Cisco IR1000 Rugged Series Secure Routers
- Cisco Catalyst IR1800 Rugged Series Routers,
- Cisco Catalyst IR8140 Heavy Duty Series Routers,
- Cisco Catalyst IR8340 Rugged Series Routers,
- Cisco Embedded Service 6300 Series Routers.



CHAPTER 12

Installing the Software

This reference provides the essential information required to install the software.

- Overview, on page 137
- Licensing, on page 137
- Install the software for Cisco IOS XE, on page 141
- IOS XE downgrade warning, on page 145
- ROMMON images, on page 146
- File systems, on page 146
- Auto-generated file directories and files, on page 147
- Flash storage, on page 148
- LED indicators, on page 148

Overview

Installing software on the router involves installing a consolidated package (bootable image). This consists of a bundle of subpackages (modular software units), with each subpackage controlling a different set of functions.

These are the two main methods to install the software:

- Managing and Configuring a Router to Run Using Consolidated Packages —This method allows for individual upgrade of subpackages and generally has reduced boot times compared to the method below. Use this method if you want to individually upgrade a module's software.
- Managing and Configuring a Router to Run Using Individual Packages —This simple method is similar to a typical Cisco router image installation and management that is supported across Cisco routers.

It is better to upgrade software in a planned period of maintenance when an interruption in service is acceptable. The router needs to be rebooted for a software upgrade to take effect.

There are different types of licenses for the device that are discussed in the next sections.

Licensing

This reference provides the necessary details regarding software licensing, activation, and compliance for the system.

Cisco software licensing

Cisco software licensing is a system that

- consists of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses
- enables licensed features and stores license files in the bootflash of your router, and
- pertains to consolidated packages, technology packages, or individual features.

Licensing support on ESR6300

The ESR6300 does not support the Right to Use licenses, and supports only the Specific License Reservation (SLR).

Configure consolidated packages

Configure the appropriate license level on the ESR6300 router to bring up all the subsystems that correspond to a specific license. The image-based license is enforced only at boot time.

An image-based license is used to help bring up all the subsystems that correspond to a license. This license is enforced only at boot time.

One of the following image-based licenses can be pre-installed on the ESR6300 router:

- Network-Essentials
- Network-Advantage
- HSecK9

Note All of the IOS-XE feature set may not apply to the ESR6300. Some features may not have been implemented yet, or are not appropriate for this platform.

Note Network-Essentials and Network-Advantage licenses are mutually exclusive. To configure one, remove the other using **no license boot level** <> option in the privileged exec mode. The router will need to be reloaded for the configuration to take effect.

To obtain software images for the router, go to: http://software.cisco.com/download/navigator.html

Procedure

Use the **license boot level** command to configure the license level using the appropriate license type.

Example:

```
Router# config terminal
            Router(config)# license boot level network-advantage
            Do you want to accept terms? Y
```

The image-based license is configured and will be enforced at the next boot time. The router must be reloaded for the configuration to take effect.

License installation on the ESR6300

License installation on the ESR6300 is a configuration requirement that

- requires two mandatory license parts: feature license and throughput license
- supports Network-Essentials or Network-Advantage for the feature license component, and
- provides throughput options of 50Mbps, 250Mbps, or 2Gbps.

License component details

The ESR6300 license consists of two required parts:

- Feature License
- Throughput License

The feature license is either Network-Essentials or Network-Advantage. The throughput license can be either 50Mbps, 250Mbps or 2Gbps.

Note For the 2Gbps throughput level, an additional feature license, HSECk9 is required to comply with export restriction requirements.

Network-Essentials

The **Network-Essentials** technology package includes the baseline features. It also supports security features.

Features that are available in this license are:

- Traffic segmentation (VPN, VRF, VLAN)
- Crypto Tunnels
- IPSec
- IKEv2
- SSL-VPN
- DHCP
- QoS
- ACL

- EIGRP
- IGMP
- HTTP
- IP Multicast
- Radius
- TACACS
- OSPF
- RIP
- HSRP

Table 15: Network-Essentials license PIDs and throughput levels

PID	Throughput Level
SL-6300-NE/DEF-K9	50Mbps
SL-6300-NE/PERF-K9	250Mbps
SL-6300-NE/BOOS-K9	2Gbps (boosted throughput)

The **Network-Essentials_npe** technology package (npe = No Payload Encryption) includes all the features in the Network-Essentials technology package without the payload encryption functionality. This is to fulfill export restriction requirements. The Network-Essentials_npe is available only in the Network-Essentials_npe image. The difference in features between the Network-Essentials package and the Network-Essentials_npe package is therefore the set of payload encryption features such as IPsec and Secure VPN.

Network-Advantage

The **Network-Advantage** technology package includes all crypto features. The Network-Advantage license contains all of the features of the Network-Essentials license, plus additional advanced networking capabilities.

The Network-Advantage license includes these additional features beyond Network-Essentials:

- MPLS
- BFD
- RSVP
- RSRB
- SDLC
- IP SLA
- STUN
- TCP optimization
- App-aware QoS policies and troubleshooting

Table 16: Network-Advantage license PIDs and throughput levels

PID	Throughput Level
SL-6300-NA/DEF-K9	50Mbps
SL-6300-NA/PERF-K9	250Mbps
SL-6300-NA/BOOS-K9	2Gbps (boosted)

The **Network-Advantage_npe** package (npe = No Payload Encryption) includes all the features in the **Network-Advantage** technology package without the payload-encryption functionality. This is to fulfill export restriction requirements. The **Network-Advantage_npe** package is available only in the **Network-Advantage_npe** image. The difference in features between the **Network-Advantage** package and the **Network-Advantage_npe** package is therefore the set of payload-encryption-enabling features such as IPsec and Secure VPN.

Policy: feature limitations for ESR6300 device deployment

The ESR6300 device has specific feature limitations that restrict certain capabilities.

- No support for MacSec or DLEP in the release.(MQC: modular quality of service command line).
- Layer2 COS to DSCP mapping does NOT work due to no ASIC chipset support for the feature.
- Copper FE SFPs are NOT supported on ESR6300.
- Copper GE SFPs are only supported in config terminal > service internal > service unsupported-transceiver mode.
- Cisco does NOT claim IP Mobility for Ethernet support on the ESR6300.
- Auto-negotiation for 10Mbps, 100Mbps, 1000Mbps in full-duplex mode is supported. For half duplex, support is only on 10Mbps and 100Mbps.
- Booting the golden configuration from a USB is NOT supported in 17.1.1, however it is available in 17.2.1.
- Refer to the Cisco approved vendor list [AVL] for Cisco USBs. Kingston USB 3.0 works as well. Ensure the USB has a single partition and ext2, fat16 or fat32 format only.
- There is no WebUI support for Day 0 or Day 1

  For Security: No support for TLS, TrustSec, MacSec , CWS [Cloud Web Security], IDS/IPS.

Install the software for Cisco IOS XE

Install Cisco IOS XE software to enable router functionality and configuration management through different package types and boot methods.

The device can be booted from multiple resources including bootflash, flash, nvram, usbflash0, and webui. Starting with IOS-XE Release 17.2.1, support has been added for booting from USB storage devices.

To install the software, use one of the following methods:

- Managing and Configuring a Router to Run Using a Consolidated Package
- Managing and Configuring a Router to Run Using Individual Packages
- Configuring a Router to Boot the Consolidated Package via TFTP Using the boot Command

Procedure

Step 1 Use the **boot config ?** command to identify the available boot resources.

Example:

```
Router(config)# boot config ?
bootflash:  URL of the config file
flash:      URL of the config file
nvram:      URL of the config file
usbflash0:  URL of the config file
webui:      URL of the config file
```

Step 2 Use the **boot config** *resource:filename* command to configure the boot source using the appropriate command syntax.

The syntax for the boot command is: `boot config usbflash0:`*`FILE name`*

The following is an example of booting from a config file on flash:

Example:

```
Router(config)# boot config flash0:startup-config
```

For USB boot configuration, use the `usbflash0` option:

```
Router(config)# boot config usbflash0:startup-config
Router(config)#
Router# write memory
Building configuration...
[OK]
*Feb 10 10:20:11.990: %SYS-2-PRIVCFG_ENCRYPT: Successfully encrypted private config file
```

Step 3 Use the **show boot** command to verify the boot configuration settings.

Example:

```
Router# show boot
BOOT variable =
CONFIG_FILE variable = usbflash0:startup-config
BOOTLDR variable does not exist
Configuration register is 0x1820
Standby not ready to show bootvar
```

The environment variable CONFIG_FILE confirms that the startup-config is set to boot from the specified location.

The Cisco IOS XE software is successfully installed and configured to boot from the selected resource, with the configuration verified through the **show boot** command output.

Boot from USB feature summary

The boot from USB feature provides specific behaviors and characteristics for configuration file management when using USB storage devices.

- Once the CONFIG_FILE is set to a non-default value, the **nvram:startup-config** command is aliased to this new location.
- Any change made to the config file in usbflash will be reflected in nvram:startup-config as well.
- The EXEC command **erase nvram:startup-config** erases the contents of NVRAM, and deletes the file referenced by CONFIG_FILE variable.
- If the USB is unplugged after setting the **boot config usbflash0:** <*filename*> variable, then the day 0 default configuration will take effect.
- When the configuration is saved using the **copy system:running-config nvram:startup-config** command, the device saves a complete version of the configuration file to the location specified by the CONFIG_FILE environment variable, and a distilled version to NVRAM. A distilled version is one that does not contain access list information.

Boot from ROMMON

This task enables you to configure the router to boot from a USB device when using ROMMON mode.

Follow these steps to boot from the USB:

Procedure

Step 1 Use the **set CONFIG_FILE=usbflash0:***filename* command to specify the configuration file from the ROMMON prompt.

Example:

```
rommon 2 > set CONFIG_FILE=usbflash0:my_startupcfg
           rommon 3 > sync
           rommon 4 > set
           PS1=rommon ! >
           MCU_UPGRADE=SKIP
           THRPUT=
           LICENSE_BOOT_LEVEL=
           RET_2_RTS=
           MCP_STARTUP_TRACEFLAGS=00000000:00000000
           BSI=0
           RANDOM_NUM=1275114933
           BOOT=flash:Jun5_1.SSA,12
           RET_2_RCALTS=951454376
           CONFIG_FILE=usbflash0:my_startupcfg
```

Step 2 Use the ROMMON prompt to continue booting the IOS image as usual.

Cisco IOS XE release upgrade

A Cisco IOS XE release upgrade is an automatic system process that:

- checks and upgrades the ROMMON version when the device boots with Cisco IOS XE image for the first time
- verifies and updates the MCU after ROMMON upgrade completion, and
- automatically power cycles the device after installing the new ROMMON.

Upgrade process details

When the device boots up with Cisco IOS XE image for the first time, the device checks the installed version of the ROMMON, and upgrades if the system is running an older version. During the upgrade, do not power cycle the device. The system automatically power cycles the device after the new ROMMON is installed.

Once ROMMON is upgraded, the device will check if the MCU is upgraded. Once that is complete, the device will boot up with the Cisco IOS XE image as normal.

After the installation, the system will boot up with the Cisco IOS XE image as normal.



Note When the device boots up for first time and if the device requires an upgrade, the entire boot process may take several minutes. This process will be longer than a normal boot due to the ROMMON and/or MCU upgrade.

Boot process of a consolidated package

This example illustrates the boot process of a consolidated package:

```
Router# configure terminal
 Router(config)#boot system bootflash:ESR 6300-universalk9.17.1.1.SPA.bin
Router(config)#config-register 0x2102
Router(config)#exit
Router#
*Nov  7 00:07:06.784: %SYS-5-CONFIG_I: Configured from console by console
Router#
Router#show run | inc license
license udi pid ESR 6300-K9 sn FCW2150TH0F
license boot level network-advantage
Router#
Router#reload ?
  /noverify  Don't verify file signature before reload.
  /verify    Verify file signature before reload.
  at         Reload at a specific time/date
  cancel     Cancel pending reload
  in         Reload after a time interval
  pause      Pause during reload
  reason     Reload reason
  <cr>       <cr>

Router#reload /verify

System configuration has been modified. Save? [yes/no]: yes
Building configuration...

[OK]
```

```
*Nov  7 00:08:48.101: %SYS-2-PRIVCFG_ENCRYPT: Successfully encrypted private config file
Verifying file integrity of bootflash:/ESR 6300-universalk9.16.10.01.SPA.bin...........
.................................

Embedded Hash   SHA1 : B0315BDC4F545D624BB128CE0FFAA468E6EF7587
Computed Hash   SHA1 : B0315BDC4F545D624BB128CE0FFAA468E6EF7587
Starting image verification
Hash Computation:    100%Done!
Computed Hash   SHA2: 03febcc07fbeadeed664f2f5ef87f6c3
                      5b343e6f7aecdd70e50e5203909aec8f
                      3d276529d2a6af6859d4c77237f812d5
                      0da93678edc942c8874edca2d5224101

Embedded Hash   SHA2: 03febcc07fbeadeed664f2f5ef87f6c3
                      5b343e6f7aecdd70e50e5203909aec8f
                      3d276529d2a6af6859d4c77237f812d5
                      0da93678edc942c8874edca2d5224101

Digital signature successfully verified in file bootflash:/ESR
6300-universalk9.16.10.01.SPA.bin
Signature Verified

Proceed with reload? [confirm]

*Jul  9 06:43:37.910: %SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload
Command. Jul  9 14:43:59.134: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting:
process exit with reload chassis code

watchdog watchdog0: watchdog did not stop!
reboot: Restarting system


Press RETURN to get started!
```

IOS XE downgrade warning

An IOS XE downgrade warning is a system feature that

- presents a warning when issuing a **boot system flash** command followed by a file name of an image which has a version number lower than the one of the running image
- allows the downgrade operation by ignoring the warning message presented to the user
- permits booting an image with the same or higher version of the running image without warning, and
- applies only to images already loaded on the bootflash of the router using the **boot system flash**<*file_name*> CLI (excluding other sources/devices like ftp, mop, rpc, tftp, rom).

Version comparison examples

The system compares versions using these rules:

When comparing two version numbers as follows:

- 17.7.1
- 17.7.1c

The version with the letter (17.7.1c) will be considered the most updated one.

When comparing two version numbers as follows:

- 17.7.3a
- 17.7.3f

The comparison will be made taking into consideration the alphabetical order. In the case above 17.7.3f will be considered the most updated one.

ROMMON images

A ROMMON image is a software package that:

- is used by ROM Monitor (ROMMON) software on a router
- is separate from the consolidated package normally used to boot the router, and
- can be independently released and used to upgrade the router with new ROMMON software.

ROMMON image release cycle

An independent ROMMON image (software package) may occasionally be released and the router can be upgraded with the new ROMMON software. For detailed instructions, see the documentation that accompanies the ROMMON image.

Note A new version of the ROMMON image is not necessarily released at the same time as a consolidated package for a router.

File systems

This table provides a list of file systems that can be seen on the Cisco ESR6300 router.

Table 17: Router file systems

File System	Description
bootflash:	Boot flash memory file system.
flash:	Alias to the boot flash memory file system above.
cns:	Cisco Networking Services file directory.
nvram:	Router NVRAM. You can copy the startup configuration to NVRAM or from NVRAM.
obfl:	File system for Onboard Failure Logging (OBFL) files.
system:	System memory file system, which includes the running configuration.

File System	Description
tar:	Archive file system.
tmpsys:	Temporary system files file system.
usbflash0:	The Universal Serial Bus (USB) flash drive file systems. **Note** The USB flash drive file system is visible only if a USB drive is installed in the USB port.

Use the **?** help option if you find a file system that is not listed in the table above.

Auto-generated file directories and files

This section discusses the auto-generated files and directories that can be created, and how the files in these directories can be managed.

Table 18: Auto-generated files

File or Directory	Description
crashinfo files	Crashinfo files may appear in the bootflash: file system. These files provide descriptive information of a crash and may be useful for tuning or troubleshooting purposes. However, the files are not part of router operations, and can be erased without impacting the functioning of the router.
core directory	The storage area for .core files. If this directory is erased, it will automatically regenerate itself at bootup. The .core files in this directory can be erased without impacting any router functionality, but the directory itself should not be erased.
tracelogs directory	The storage area for trace files. Trace files are useful for troubleshooting. If the Cisco IOS process fails, for instance, users or troubleshooting personnel can access trace files using diagnostic mode to gather information related to the Cisco IOS failure. Trace files, however, are not a part of router operations, and can be erased without impacting the router's performance.

Important notes about auto-generated directories

Important information about auto-generated directories include:

- auto-generated files on the bootflash: directory should not be deleted, renamed, moved, or altered in any way unless directed by Cisco customer support.

Note Altering auto-generating files on the bootflash: may have unpredictable consequences for system performance.

- Crashinfo files and files in the core and tracelogs directory can be deleted.

Flash storage

Flash storage is a local media storage that enables subpackage installation and router operation.

Flash storage operation

To list file names in flash storage, use the **dir bootflash:** command.

Note Flash storage is required for successful operation of a router.

LED indicators

For LED information on the router, see the LED section of the ESR6300 Hardware Technical Manual.



CHAPTER 13

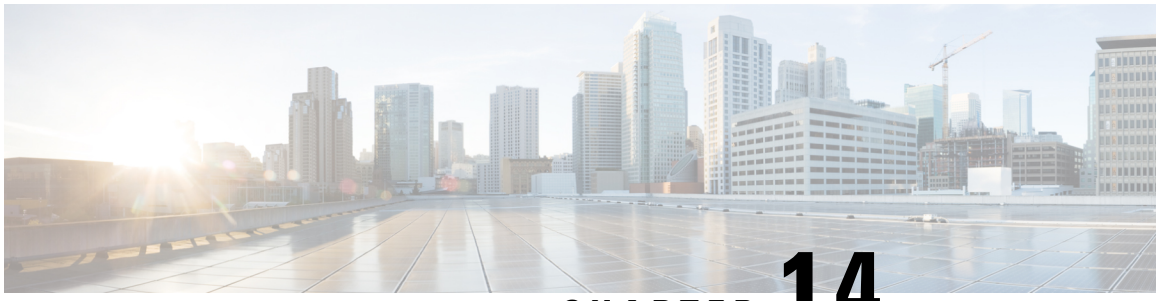
Smart Licensing Using Policy

This reference provides information regarding the implementation and management of Smart Licensing Using Policy.

- Smart licensing using policy on Cisco routers, on page 149

Smart licensing using policy on Cisco routers

The Smart Licensing Using Policy chapter has been replaced by a standalone guide called Smart Licensing Using Policy on the Cisco Catalyst IR1101, IR1800, IR8140, IR8340 and ESR6300 Routers.



CHAPTER 14

Configuring Ethernet Switch Ports

This section provides the necessary information for configuring Ethernet switch ports.

- Configuring VLANs, on page 151
- VLAN trunking protocol (VTP), on page 153
- 802.1x authentication, on page 153
- Configure spanning tree protocol, on page 154
- MAC address table manipulation, on page 157
- Configure switch port analyzer, on page 158
- IGMP snooping, on page 159

Configuring VLANs

A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment. Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in the VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router.

On the ESR6300, all the Gigabit Ethernet LAN ports g0/1/0 through g0/1/4 are set up in vlan1, which does not need to be created.

Note ESR6300 routers have reserved a set of VLANs (2350 to 2449) for additional usage. It is not allowed to add the reserved VLANs. You must ensure that these VLANs are not used in the network.

Procedure

Step 1 Use the **switchport access vlan** *vlan-id* command to assign a port to a VLAN.

Example:

```
Router# configure terminal
Router(config)# interface g 0/1/0
Router(config-if)# switchport access vlan 4
```

```
Router(config-if)# no shut
Router(config-if)# exit
Router(config)# interface vlan 4
Router(config-if)# ipv4 address {ip} {mask}
Router(config-if)# ipv6 enable
Router(config-if)# ipv6 address autoconfig
Router(config-if)# no shut
Router(config-if)# end
```

Step 2 Use the **show run interface g0/1/0**, **show interface g0/1/0**, and **show vlan** commands to verify the configuration.

The port is assigned to the specified VLAN, and the VLAN interface is configured with the assigned IP addressing. The configuration can be confirmed using the **show run interface**, **show interface**, and **show vlan** command outputs.

This is an example of a VLAN configuration:

```
Router# show vlan
        VLAN  Name                             Status       Ports
        ---- -------------------------------- --------- -------------------------------
        1     default                          active       Gi0/1/0, Gi0/1/1, Gi0/1/2 Gi0/1/3

        1002  fddi-default                      act/unsup
        1003  token-ring-default                act/unsup
        1004  fddinet-default                   act/unsup
        1005  trnet-default                     act/unsup

        VLAN  Type  SAID      MTU   Parent  RingNo BridgeNo Stp BrdgMode Trans1 Trans2
        ---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
        1     enet  100001     1500  -       -      -       -    -         0      0
        1002  fddi  101002     1500  -       -      -       -    -         0      0
        1003  tr    101003     1500  -       -      -       -    -         0      0
        1004  fdnet 101004     1500  -       -      -       ieee -         0      0
        1005  trnet 101005     1500  -       -      -       ibm  -         0      0

        Primary Secondary Type Ports
        ------- --------- ----------------- ------------------------------------------
        Router#
```

What to do next

IOS-XE supports Embedded Packet Capture (EPC), which provides an embedded systems management facility that helps in tracing and troubleshooting packets. This feature allows network administrators to capture data packets flowing through, to, and from a Cisco device.

The network administrator may define the capture buffer size and type (circular or linear), the maximum number of bytes of each packet to capture, and the direction of the traffic flow—ingress or egress, or both. The packet capture rate can be throttled using further administrative controls.

For example, you can use the available options for filtering the packets to be captured using an Access Control List and, optionally, further define the capture by specifying a maximum packet capture rate or by specifying a sampling interval. For additional details, see Embedded Packet Capture Configuration Guide, Cisco IOS XE Gibraltar 16.10.x.

VLAN trunking protocol (VTP)

VTP is a Layer 2 messaging protocol that

- maintains VLAN configuration consistency by managing the addition, deletion, and renaming of VLANs on a network-wide basis
- minimizes mis-configurations and configuration inconsistencies that can cause problems such as duplicate VLAN names, incorrect VLAN-type specifications, and security violations, and
- enables centralized configuration changes on one or more switches with automatic communication to all other switches in the network.

VTP operation characteristics

Before you create VLANs, you must decide whether to use VTP in your network. Using VTP, you can make configuration changes centrally on one or more switches and have those changes automatically communicated to all the other switches in the network. Without VTP, you cannot send information about VLANs to other switches.

VTP is designed to work in an environment where updates are made on a single switch and are sent through VTP to other switches in the domain. It does not work well in a situation where multiple updates to the VLAN database occur simultaneously on switches in the same domain, which would result in an inconsistency in the VLAN database.

Further information about configuring VTP can be found here: http://www.cisco.com/c/en/us/td/docs/routers/access/interfaces/software/feature/guide/geshwic_cfg.html#wp1046901

802.1x authentication

IEEE 802.1x port-based authentication is a client-server-based access control and authentication protocol that:

- prevents unauthorized clients from connecting to a LAN through publicly accessible ports,
- requires authentication server validation before allowing access to switch or LAN services, and
- allows only Extensible Authentication Protocol over LAN (EAPOL), Cisco Discovery Protocol (CDP), and Spanning Tree Protocol (STP) traffic through the port until the client is authenticated.

Device roles in 802.1x authentication

With IEEE 802.1x authentication, the devices in the network have specific roles:

- Supplicant—Device (workstation) that requests access to the LAN and switch services and responds to requests from the router. The workstation must be running IEEE 802.1x-compliant client software such as that offered in the Microsoft Windows XP operating system. (The supplicant is sometimes called the client.)
- Authentication server—Device that performs the actual authentication of the supplicant. The authentication server validates the identity of the supplicant and notifies the router whether or not the supplicant is authorized to access the LAN and switch services. The Network Access Device transparently passes the authentication messages between the supplicant and the authentication server, and the authentication

process is carried out between the supplicant and the authentication server. The particular EAP method used will be decided between the supplicant and the authentication server (RADIUS server). The RADIUS security system with EAP extensions is available in Cisco Secure Access Control Server Version 3.0 or later. RADIUS operates in a client and server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.

- Authenticator—Router that controls the physical access to the network based on the authentication status of the supplicant. The router acts as an intermediary between the supplicant and the authentication server, requesting identity information from the supplicant, verifying that information with the authentication server, and relaying a response to the supplicant. The router includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.



Note The ESR6300 supports authentication/authorization from TACACS+ server as well.

For detailed information on how to configure 802.1x port-based authentication, see this link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_8021x/configuration/15-mt/sec-user-8021x-15-mt-book/config-ieee-802x-pba.html

Enabling IEEE 802.1x and AAA on a switch port

This example shows how to configure an ESR6300 router as 802.1x authenticator:

```
Router> enable
Router# configure terminal
Router(config)# dot1x system-auth-control
Router(config)# aaa new-model
Router(config)# aaa authentication dot1x default group radius
Router(config)# interface GigabitEthernet0/1/0
Router(config-if)# switchport mode access
Router(config-if)# access-session port-control auto
Router(config-if)# dot1x pae authenticator
Router(config-if)# access-session closed
Router(config-if)# access-session host-mode single-host
Router(config-if)# end
```

Configure spanning tree protocol

Spanning Tree Protocol (STP) is configured to manage Layer 2 links by providing path redundancy while preventing network loops.

Spanning Tree Protocol (STP) is a Layer 2 link management protocol that provides path redundancy while preventing loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. Multiple active paths among end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages. Switches might also learn end-station MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network. Spanning-tree operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

The STP uses a spanning-tree algorithm to select one switch of a redundantly connected network as the root of the spanning tree. The algorithm calculates the best loop-free path through a switched Layer 2 network by assigning a role to each port based on the role of the port in the active topology:

- Root—A forwarding port elected for the spanning-tree topology
- Designated—A forwarding port elected for every switched LAN segment
- Alternate—A blocked port providing an alternate path to the root bridge in the spanning tree
- Backup—A blocked port in a loopback configuration

The switch that has all of its ports as the designated role or as the backup role is the root switch. The switch that has at least one of its ports in the designated role is called the designated switch. Spanning tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the spanning-tree algorithm recalculates the spanning-tree topology and activates the standby path. Switches send and receive spanning-tree frames, called bridge protocol data units (BPDUs), at regular intervals. The switches do not forward these frames but use them to construct a loop-free path. BPDUs contain information about the sending switch and its ports, including switch and MAC addresses, switch priority, port priority, and path cost. Spanning tree uses this information to elect the root switch and root port for the switched network and the root port and designated port for each switched segment.

When two ports on a switch are part of a loop, the spanning-tree port priority and path cost settings control which port is put in the forwarding state and which is put in the blocking state. The spanning-tree port priority value represents the location of a port in the network topology and how well it is located to pass traffic. The path cost value represents the media speed.

For detailed configuration information on STP, see: STP Configuration Guide.

Procedure

Step 1 Use the **spanning-tree vlan** *vlan-id* **port-priority** command to configure spanning-tree port priority for a Gigabit Ethernet interface.

Example:

```
Router# configure terminal
Router(config)# interface GigabitEthernet0/1/0
Router(config-if)# spanning-tree vlan 1 port-priority 64
Router(config-if)# end
```

If a loop occurs, spanning tree uses the port priority when selecting an interface to put in the forwarding state.

Step 2 Use the **spanning-tree cost** command to change the spanning-tree port cost for a Gigabit Ethernet interface.

Example:

```
Router# configure terminal
Router(config)# interface GigabitEthernet0/1/0
Router(config-if)# spanning-tree cost 18
Router(config-if)# end
```

If a loop occurs, spanning tree uses cost when selecting an interface to put in the forwarding state.

Step 3 Use the **spanning-tree vlan** *vlan-id* **priority** command to configure the bridge priority for a VLAN.

Example:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 priority 33792
Router(config)# end
```

Step 4 Use the **spanning-tree vlan** *vlan-id* **hello-time** command to configure the hello time for a VLAN.

Example:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 hello-time 7
Router(config)# end
```

The hello time is the interval between the generation of configuration messages by the root switch.

Step 5 Use the **spanning-tree vlan** *vlan-id* **forward-time** command to configure the forward delay time for a VLAN.

Example:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 forward-time 21
Router(config)# end
```

The forward delay is the number of seconds an interface waits before changing from its spanning-tree learning and listening states to the forwarding state.

Step 6 Use the **spanning-tree vlan** *vlan-id* **max-age** command to configure the maximum age interval for the spanning tree.

Example:

```
Router# configure terminal
Router(config)# spanning-tree vlan 20 max-age 36
Router(config)# end
```

The maximum-aging time is the number of seconds a switch waits without receiving spanning-tree configuration messages before attempting a reconfiguration.

Step 7 Use the **spanning-tree vlan** *vlan-id* **root primary diameter** command to configure the switch as the root bridge for a VLAN with the specified network diameter.

Example:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 root primary diameter 4
Router(config)# exit
```

Spanning Tree Protocol is configured with the specified parameters for port priority, cost, bridge priority, timing intervals, and root bridge designation to manage Layer 2 network topology and prevent loops.

Example

The example shows configuring the bridge priority of VLAN 10 to 33792:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 priority 33792
Router(config)# end
```

The example shows configuring the hello time for VLAN 10 being configured to 7 seconds:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 hello-time 7
Router(config)# end
```

The example shows configuring forward delay time:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 forward-time 21
Router(config)# end
```

The example shows the switch being configured as the root bridge for VLAN 10, with a network diameter of 4:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 root primary diameter 4
Router(config)# exit
```

MAC address table manipulation

MAC address table manipulation is a network administration process that

- configures address information that the switch uses to forward traffic between ports
- manages both dynamic and static MAC address entries in the forwarding table, and
- controls aging time settings to define retention periods for unseen addresses.

MAC address types

The MAC address table contains these types of addresses:

- Dynamic address: a source MAC address that the switch learns and then drops when it is not in use. You can use the aging time setting to define how long the switch retains unseen addresses in the table.
- Static address: a manually entered unicast address that does not age and that is not lost when the switch resets.

The address table lists the destination MAC address, the associated VLAN ID, and port associated with the address and the type (static or dynamic).

For detailed configuration information on MAC address table manipulation see this link:

http://www.cisco.com/c/en/us/td/docs/routers/access/interfaces/software/feature/guide/geshwic_cfg.html#wp1048223

Example for MAC address table manipulation

This example shows creating a static entry in the MAC address table:

```
Router# configure terminal
Router(config)# mac address-table static 0002.0003.0004 interface GigabitEthernet0/1/0 vlan
 3
Router(config)# end
```

This example shows setting the aging timer:

```
Router# configure terminal
Router(config)# mac address-table aging-time 300
Router(config)# end
```

Configure switch port analyzer

Configure Switch Port Analyzer to analyze network traffic passing through ports by sending a copy of the traffic to another port connected to a network analyzer or other monitoring or security device.

The Cisco ESR6300 supports local SPAN only, and up to one SPAN session. SPAN copies (or mirrors) traffic received or sent (or both) on source ports to a destination port for analysis. SPAN does not affect the switching of network traffic on the source ports. You must dedicate the destination port for SPAN use. Except for traffic that is required for the SPAN or RSPAN session, destination ports do not receive or forward traffic.

Only traffic that enters or leaves source ports or traffic that enters or leaves source can be monitored by using SPAN; traffic routed to a source cannot be monitored. For example, if incoming traffic is being monitored, traffic that gets routed from another source cannot be monitored; however, traffic that is received on the source and routed to another can be monitored.

For detailed information on how to configure a switched port analyzer (SPAN) session, see the following web link: SPAN Configuration Guide.

Procedure

Step 1 Use the **monitor session** *session-number* **source** command to configure a SPAN session to monitor bidirectional traffic from an interface.

Example:

```
Router# configure terminal
            Router(config)# monitor session 1 source GigabitEthernet0/1/0
            Router(config)# end
```

Step 2 Use the **monitor session** *session-number* **destination** command to configure an interface as the destination for a SPAN session.

Example:

```
Router# configure terminal
            Router(config)# monitor session 1 destination GigabitEthernet0/1/0
            Router(config)# end
```

Step 3 Use the **no monitor session** *session-number* **source** command to remove the interface as a SPAN source for SPAN session 1 if needed.

Example:

```
Router# configure terminal
            Router(config)# no monitor session 1 source GigabitEthernet0/1/0
            Router(config)# end
```

The SPAN session is configured to monitor network traffic from the specified source interface and copy it to the destination interface for analysis.

IGMP snooping

IGMP snooping is a Layer 2 switching feature that

- constrains the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices
- requires the LAN switch to snoop on the IGMP transmissions between the host and the router and to keep track of multicast groups and member ports, and
- adds and removes host ports from forwarding table entries based on IGMP report and Leave Group messages, while periodically deleting entries if it does not receive IGMP membership reports from multicast clients.

IGMP snooping operation

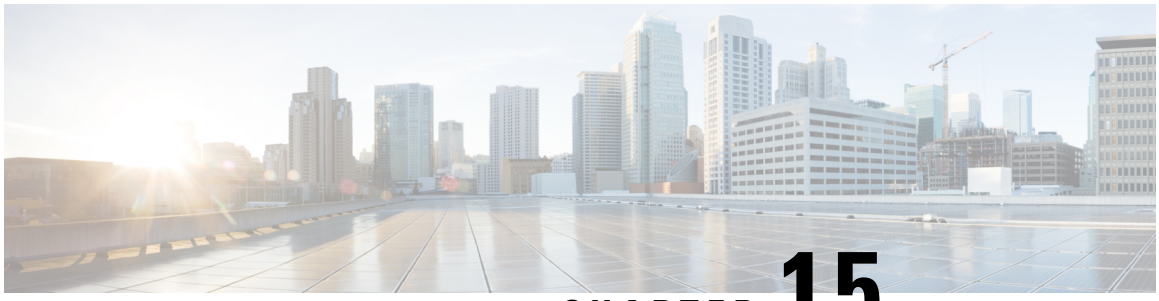
The multicast router sends out periodic general queries to all VLANs. All hosts interested in this multicast traffic send join requests and are added to the forwarding table entry.

When the switch receives an IGMP report from a host for a particular multicast group, the switch adds the host port number to the forwarding table entry; when it receives an IGMP Leave Group message from a host, it removes the host port from the table entry.

Use the **IP IGMP snooping enable** command to configure IGMP Snooping on the ESR6300.

By default, IGMP snooping is globally enabled in the ESR6300.

MLD snooping is also supported on the ESR6300, and further information can be found in this documentation set: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3850/software/release/16-1/configuration_guide/b_161_consolidated_3850_cg/b_161_consolidated_3850_cg_chapter_01100.html



CHAPTER 15

Gigabit Ethernet Combo Ports WAN

This section contains the following topics:

- Gigabit ethernet combo ports WAN, on page 161
- Configure the WAN port media type, on page 162

Gigabit ethernet combo ports WAN

A Gigabit Ethernet combo port is a dual-connectivity interface that

- combines both copper RJ45 and fiber SFP connectivity options in paired ports
- allows only one port in each pair to be active at any given time, and
- automatically selects the first port that establishes a valid link as the active connection.

ESR6300 combo port specifications

Two combo ports are available on the ESR6300 with interface names g0/0/0 and g0/0/1. The copper ports support standard 3-speed, 10/100/1000Base-T protocol. The SFP supports standard 1000Base-X or 100Base-FX Ethernet over single-mode or multi-mode optics. A complete list of the supported SFPs can be found in the ESR6300 Hardware Technical Manual: https://www.cisco.com/c/en/us/td/docs/routers/embedded/6300/hardware/tech-guide/ESR6300-tech-guide.html.

Note The GLC-T and GLC-TE are the only copper SFPs supported by Cisco, and the **service unsupported-transceiver** CLI must be manditorily enabled if they are used.

Configuration considerations include these requirements:

- Copper FE SFPs are not supported on ESR6300
- Auto-negotiation for 10Mbps, 100Mbps, 1000Mbps in full-duplex and half-duplex mode is supported. For half duplex, support is only on 10Mbps and 100Mbps.
- Auto-select is supported. Whichever interface is detected first is selected. Typically fiber SFP bring-UP is faster than RJ45 Copper is faster than Copper SFP.
- All Standard Cisco Layer 3 Routing Protocols are Supported on ESR6300 WAN interfaces.

- If running IOS-XE version 17.1.x and below, Jumbo MTU support is not available on the ESR6300. MTU >1500 packets will be fragmented. If running IOS-XE release 17.2.1 and above, MTU support is 64 - 9216 on the interface. If there are bursts of fragmented traffic expected beyond the device Non-drop rate capability, IOS-XE implements a throttling mechanism and transmits MAC pause frames to other end to reduce rate of traffic. To resume normal operation, ARP has to be sent again to re-negotiate. The default ip ARP timeout is 14,000 seconds. In such corner case scenarios, it is recommended to have a much shorter ARP timeout configuration.
- Auto-select is supported. Whichever interface is detected first is selected. Typically, fiber SFP bring-UP is faster than RJ45 Copper, which is faster than Copper SFP.
- Auto-failover is supported as well between like GE Ethernet and GE SFP. In case one physical link goes DOWN, the other mode will immediately kick-in.
- RJ45 to SFP mode conversion involves an interface reset (shut/no shut, interface DOWN/UP). The syslog can be observed accordingly.



Note For ARP, you must configure Ethernet II frames instead of 802.3 frames. If 802.3 frames are used, they may be recorded as SNAP encapsulation, which is currently not supported.

Configure the WAN port media type

Use this task to configure a WAN port as RJ45 or SFP using the command line interface and to verify the configuration. Configuring WAN port media types optimizes network connectivity and enables failover capabilities between RJ45 and SFP connections.

WAN ports can be configured to use either RJ45 or SFP connectors, with auto-failover capabilities available. The system supports auto-selection of the attached connector type or manual configuration of specific media types.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# config terminal
```

Step 2 Use the **interface** *type* **g0/0/***{0|1}* command to select the WAN interface.

Example:

```
Router(config)# interface g0/0/{0|1}
```

Step 3 Use the **media-type** *type* command to set the media type. The following options are available:

Example:

```
Router(config-if)# media-type ?
auto-select  Use whichever connector is attached
rj45         Use RJ45 connector
sfp          Use SFP connector
```

Step 4 (Optional) Use the **media-type** *{rj45|sfp}* **auto-failover** command to configure auto-failover.

Example:

```
Router(config-if)# media-type {rj45|sfp} ?
auto-failover  Automatic Fail over
```

Step 5 Use these steps to change the media type to SFP:

a) Use the **interface** *type* **g 0/0/0** command to select the interface.

Example:

```
Router# conf t
Router(config)# int g 0/0/0
```

b) Use the **no shutdown** command to enable the interface.

Example:

```
Router(config-if)# no shut
```

c) Use the **media-type sfp** command to change the media type to SFP.

Example:

```
Router(config-if)# media-type sfp
GigabitEthernet0/0/0: Changing media to SFP
Router(config-if)# end
```

Step 6 Use the **show inventory** command to verify that the SFP is detected.

Example:

```
Router# show inventory

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INFO: Please use "show license UDI" to get serial number for licensing.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

NAME: "Chassis", DESCR: "Embedded Services Router with 2 Gigabit Ethernet routed ports,4
Gigabit Ethernet switched ports, no cooling plate"
PID: ESR-6300-NCP-K9 , VID: V00 , SN: FOC23032UUN

NAME: "Power Supply Module 0", DESCR: "External Power Supply Module"
PID: PWR-12V , VID: , SN:

NAME: "module 0", DESCR: "Cisco ESR-6300-NCP-K9 Fixed and pluggable Interface Module
controller"
PID: ESR-6300-NCP-K9 , VID: , SN:

NAME: "NIM subslot 0/1", DESCR: "ESR-6300-ES-4"
PID: ESR-6300-ES-4 , VID: V01 , SN:

NAME: "NIM subslot 0/0", DESCR: "Front Panel 2 port Gigabitethernet Module"
PID: ESR-6300-2x1GE , VID: V01 , SN:

NAME: "module R0", DESCR: "Cisco ESR-6300-NCP-K9 Route Processor"
PID: ESR-6300-NCP-K9 , VID: V00 , SN: FOC23032UUN

NAME: "module F0", DESCR: "Cisco ESR-6300-NCP-K9 Forwarding Processor"
PID: ESR-6300-NCP-K9 , VID: , SN:
```

Step 7 Verify that the configuration has taken effect by reapplying the media-type configuration on the interface.

Example:

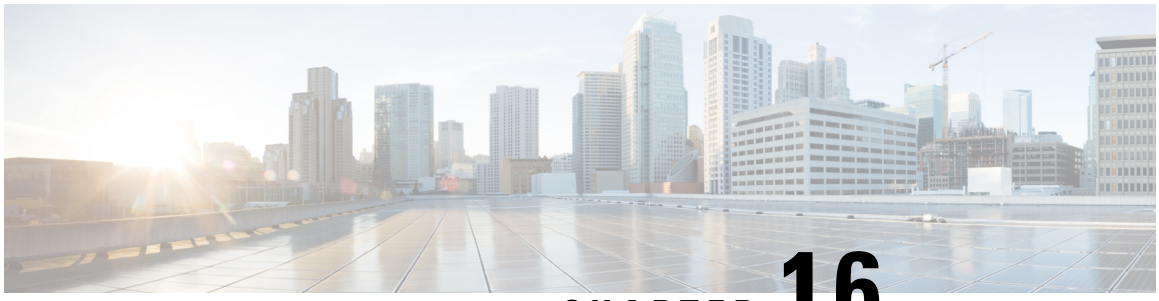
```
Router# conf t
Router(config)# int g 0/0/0
Router(config-if)# no shut
Router(config-if)# media-type sfp
GigabitEthernet0/0/0: Changing media to SFP
Router(config-if)# end
```

Step 8 (Optional) Use the **hw-module subslot 0/0 reload force** command to reload the Gigabit Ethernet module if necessary.

Example:

```
Router# hw-module subslot 0/0 reload force
```

The WAN port is successfully configured with the specified media type. The system displays confirmation messages when changing media types, and the configuration can be verified using inventory and interface commands.



CHAPTER 16

SD-WAN Remote Access

- SD-WAN, on page 165

SD-WAN

SD-WAN RA is a networking solution that

- combines IOS-XE SD-WAN and IOS-XE FlexVPN Remote Access Server features
- is supported on IoT routers with IOS XE 17.13.1, and
- provides remote access capabilities where all IoT devices only support the SD-WAN RA Client.

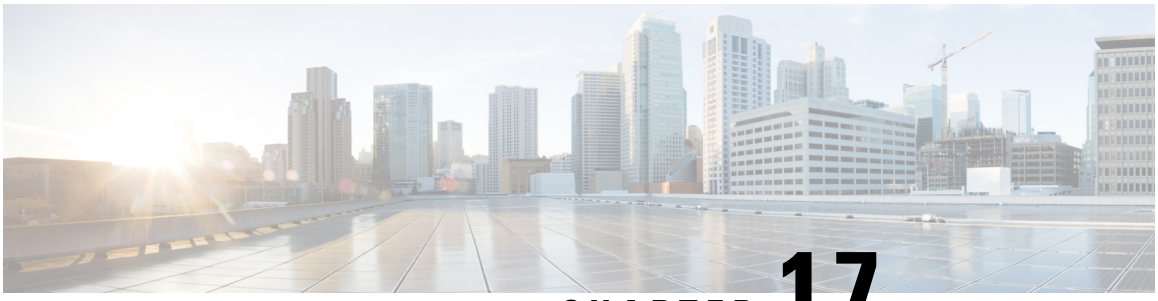
Additional documentation

Note All IoT devices only support the SD-WAN RA Client.

Information on SD-WAN Remote Access can be found in Cisco Catalyst SD-WAN Remote Access.

Additional documentation for SDWAN/vManage is available at these links:

- User Documentation for Cisco IOS XE Catalyst SD-WAN Release 17
- Cisco Catalyst SD-WAN
- Cisco SD-WAN Support Information
- Cisco vManage Monitor Overview
- Managing the SD-Routing Device Using Cisco SD-WAN Manager



CHAPTER 17

CLI Output for the FN980 5G Modem

• Change in CLI output for the FN980 5G modem, on page 167

Change in CLI output for the FN980 5G modem

This release has a different output to the **show cellular 0/x/0 radio band** command. The module will no longer display the 5G-SA band information by default. However, once the 5G-SA has been enabled, the band information will then be displayed.

Command examples

See these command examples using a router running IOS XE 17.13.1 with an FN980 modem.

```
router# show cellular 0/3/0 radio band
LTE bands supported by modem:
- Bands 2 4 5 12 14 26 29 30 46 48 66.
LTE band Preference settings for the active sim(slot 0):
- Bands 2 4 5 12 14 26 29 30 46 48 66.

NR5G NSA bands supported by modem:
- Bands 2 5 12 66 77.
NR5G NSA band Preference settings for the active sim(slot 0):
- Bands 2 5 12 66 77.

3G bands supported by modem:
Index: <none>
3G band Preference settings for the active sim(slot 0):
Index: <none>

==========================================
Band index reference list:

For LTE and 5G, indices 1-128 correspond to bands 1-128.

For 3G, indices 1-64 maps to the 3G bands mentioned against each above.
```

```
router#show cellular 0/3/0 radio hard
*Nov  8 12:13:31.969: Graphit 5G RSRP/RSRQ LTE modem:[1]
Modem Firmware Version = M0H.030202
Host Firmware Version = A0H.000302
Device Model ID =  FN980
International Mobile Subscriber Identity (IMSI) = 001010123456789
International Mobile Equipment Identity (IMEI) = 359661100035795
Integrated Circuit Card ID (ICCID) = 898600005020001807 22
Mobile Subscriber Integrated Services
```

```
Digital Network-Number (MSISDN) =
Modem Status = Modem Online
Current Modem Temperature = 40 deg C
PRI version = 1080-114, Carrier = Generic GCF
OEM PRI version = 1080-114

router# show cellular 0/3/0 radio band
LTE bands supported by modem:
- Bands 1 2 3 4 5 7 8 12 13 14 17 18 19 20 25 26 28 29 30 32 34 38 39 40 41 42 43 46 48 66
 71.
LTE band Preference settings for the active sim(slot 0):
- Bands 1 2 3 4 5 7 8 12 13 14 17 18 19 20 25 26 28 29 30 32 34 38 39 40 41 42 43 46 48 66
 71.

NR5G NSA bands supported by modem:
- Bands 1 2 3 5 7 8 12 20 25 28 38 40 41 48 66 71 77 78 79.
NR5G NSA band Preference settings for the active sim(slot 0):
- Bands 1 2 3 5 7 8 12 20 25 28 38 40 41 48 66 71 77 78 79.

NR5G SA bands supported by modem:
- Bands <none>
NR5G SA band Preference settings for the active sim(slot 0):
- Bands <none>

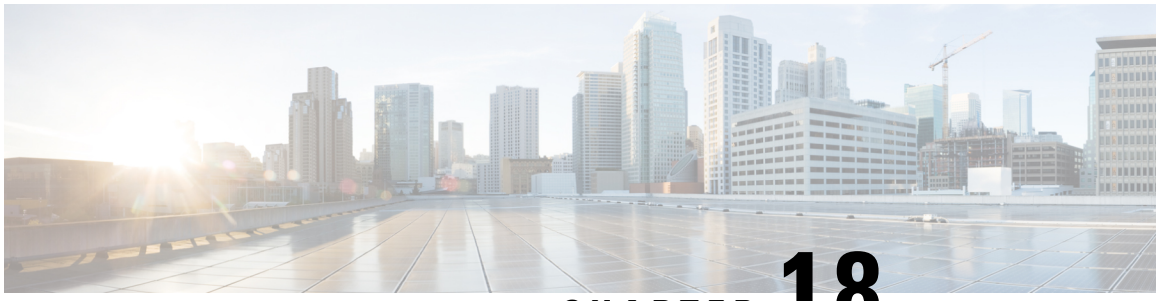
3G bands supported by modem:
Index:
  23 - UMTS Band 1: 2100 MHz (IMT)
  24 - UMTS Band 2: 1900 MHz (PCS A-F)
  26 - UMTS Band 4: 1700 MHz (AWS A-F)
  27 - UMTS Band 5: US 850 MHz  (CLR)
  50 - UMTS Band 8: 900 MHz  (E-GSM)
  51 - UMTS Band 9: Japan 1700 MHz
  61 - UMTS Band 19: 800 MHz (800 Japan)
3G band Preference settings for the active sim(slot 0):
Index:
  23 - UMTS Band 1: 2100 MHz (IMT)
  24 - UMTS Band 2: 1900 MHz (PCS A-F)
  26 - UMTS Band 4: 1700 MHz (AWS A-F)
  27 - UMTS Band 5: US 850 MHz  (CLR)
  50 - UMTS Band 8: 900 MHz  (E-GSM)
  51 - UMTS Band 9: Japan 1700 MHz
  61 - UMTS Band 19: 800 MHz (800 Japan)

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

Band index reference list:

For LTE and 5G, indices 1-128 correspond to bands 1-128.

For 3G, indices 1-64 maps to the 3G bands mentioned against each above.
```



CHAPTER 18

MAB 802-1x Support

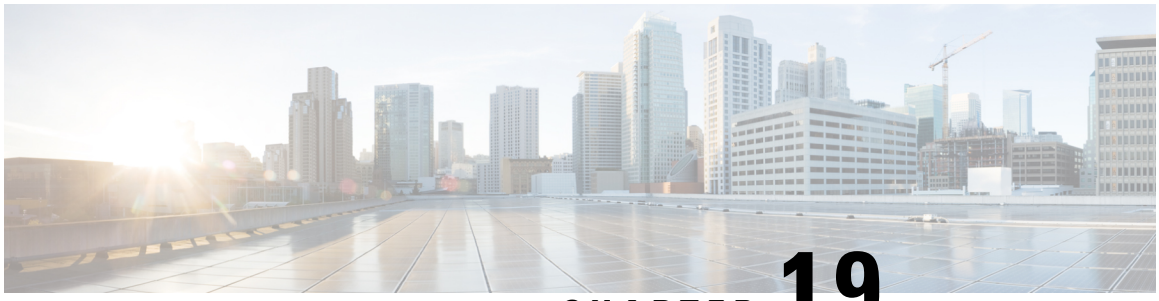
- MAB 802.1x support, on page 169

MAB 802.1x support

MAB 802.1x support is an authentication method that grants network access to specific MAC addresses regardless of 802.1X capability or credentials, allowing devices such as cash registers, fax machines, and printers to be readily authenticated and making network features that are based on authorization policies available.

Additional information

For further information, see the Authentication Authorization and Accounting Configuration Guide, Cisco IOS XE



CHAPTER 19

Enable Secure Data Wipe Capabilities

- Enable the secure data wipe capability, on page 171

Enable the secure data wipe capability

Use this task to configure secure data wipe on IoT platforms to ensure storage devices are properly purged using NIST SP 800-88r1 compliant secure erase commands.

Secure data wipe is a Cisco-wide initiative to ensure that storage devices on all IOS XE based platforms are properly purged using NIST SP 800-88r1 compliant secure erase commands. Whenever possible, IoT platforms leverage the corresponding ENG design and implementation available on their platforms.

This feature is supported on the following IoT platforms:

- IR1101
- IR1800
- IR8140
- ESR6300

When secure data wipe is executed, the following are wiped out:

- IR1101, IR1800, IR8140: NVRAM, rommon variables, and bootflash.
- ESR6300: NVRAM, rommon variables, and bootflash.

Procedure

Step 1 Use the **factory-reset all secure** command to execute the secure data wipe. When prompted to confirm, enter **Y**.

Example:

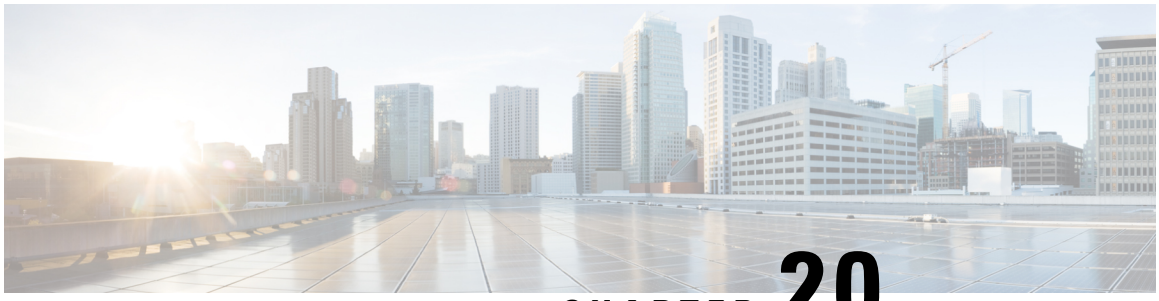
```
Router# factory-reset all secure
The factory reset operation is irreversible for securely reset all. Are you sure? [confirm]Y
This operation may take hours. Please do not power cycle.
```

Step 2 Use the **show platform software factory-reset secure log** command after the command is executed and IOS XE boots up, to check the log.

Example:

```
Router# show platform software factory-reset secure log
Factory reset log:
#CISCO DATA SANITIZATION REPORT:# IR1800
Purge ACT2 chip at 12-08-2022, 15:17:28
ACT2 chip Purge done at 12-08-2022, 15:17:29
mtd and backup flash wipe start at 12-08-2022, 15:17:29
mtd and backup flash wipe done at 12-08-2022, 15:17:29.
```

The router is in the rommon prompt with default factory settings (baud rate 9600) after the command is executed. The bootflash is not formatted until booting with an IOS image through usbflash or TFTP download, if the platform is supported.



CHAPTER 20

Implementing Multicast Service Reflection

This chapter contains the following topics:

- Multicast service reflection, on page 173
- Configure PIMv6 Anycast RP solution, on page 186

Multicast service reflection

Multicast service reflection is a Cisco networking feature that

- provides the capability for users to translate externally received multicast or unicast destination addresses to multicast or unicast addresses that conform to their organization's internal addressing policy
- eliminates the need to redistribute unicast routes from external sources at the translation boundary into network infrastructure for Reverse Path Forwarding (RPF) to work properly, and
- allows users to receive identical feeds from two ingress points in the network and route them independently.

Prerequisites for implementing multicast service reflection

Complete these prerequisites before implementing multicast service reflection:

- Configure your multicast-enabled network with the necessary infrastructure to run either Protocol Independent Multicast Sparse Mode (PIM-SM), Bidirectional PIM (bidir-PIM), or PIM Source Specific Multicast (PIM-SSM). The configuration process may include configuring RPs, interface boundaries, or SSM ranges.
- Confirm that the virtual interface for multicast service reflection (Vif1 interface) is installed in your border router and the Multicast Service Reflection application is installed and operational.
- Each active receiver must initiate an Internet Group Management Protocol (IGMP) join to the multicast group that is defined on the router in the PIM domain.

Restrictions for implementing multicast service reflection

Be aware of these restrictions when implementing multicast service reflection:

- When translating groups of multicast packets that are destined for the same multicast group but are originating from different sources, as in the case when using PIM-SSM, all multicast packets destined for a particular SSM group will get mapped to a single (S, G) after translation has occurred. For example, if (10.1.1.1, 232.1.1.1) and (10.1.1.2, 232.1.1.1) need to be translated, they will appear as a single entry, for example, (92.168.1.2, 232.239.1.1), where 192.168.1.2 is an IP address that resides in the Vif1 IP subnet.
- PIM/IGMP control packets are not translated.
- The following types of translation are not supported in Cisco IOS XE software:
  - Unicast-to-Multicast Destination Translation
  - Unicast-to-Multicast Destination Splitting
- The use of a sub-interface as an input-interface has not been tested on platforms that are based on mcp_dev.

Information about implementing multicast service reflection

This reference provides the necessary configuration requirements and operational considerations for implementing multicast service reflection in network environments.

Benefits of implementing multicast service reflection

Benefits of implementing multicast service reflection are networking advantages that

- allow users to translate externally received multicast or unicast destination addresses to multicast or unicast addresses that conform to their company's internal addressing policy,
- provide logical separation between private and public multicast networks, and
- provide flexibility to forward multicast packets--translated or untranslated--out the same outgoing interface.

Additional multicast service reflection benefits

Multicast service reflection offers these additional capabilities:

- Provides redundancy by allowing users to get identical feeds from two ingress points in the network and route them independently.
- Allows users to use the subnet of their choice to be the source network and scope it appropriately.

The address translation capability supports these types of translations:

- Multicast-to-Multicast Destination Translation
- Multicast-to-Unicast Destination Translation
- Multicast-to-Multicast Destination Splitting
- Multicast-to-Unicast Destination Splitting

Rendezvous points

A Rendezvous Point (RP) is a role that a router performs that

- operates in PIM-SM or bidirectional PIM
- acts as the meeting place for sources and receivers of multicast data, and
- enables forwarding of traffic only to network segments with active receivers that have explicitly requested multicast data in PIM-SM.

RP operation in PIM-SM networks

In a PIM-SM network, the RP facilitates multicast distribution through this process:

1. First hop designated routers with directly connected sources initially send their traffic to the RP.
2. This traffic is then forwarded to receivers down a shared distribution tree.
3. By default, when the last hop router with a directly connected receiver receives traffic from the shared tree, it immediately performs a shortest path tree switchover and sends a Join message towards the source, creating a source-based distribution tree between the source and the receiver.



Note An RP is required only in networks running PIM-SM or bidirectional PIM.

PIM sparse mode

PIM sparse mode (PIM-SM) is a multicast routing protocol that

- uses a pull model to deliver multicast traffic
- delivers traffic only to network segments with active receivers that have explicitly requested the data
- requires the use of a rendezvous point (RP) for shared tree operation

PIM sparse mode operation

Unlike dense mode interfaces, sparse mode interfaces are added to the multicast routing table only when periodic Join messages are received from downstream routers, or when a directly connected member is on the interface. When forwarding from a LAN, sparse mode operation occurs if an RP is known for the group. If so, the packets are encapsulated and sent toward the RP. When no RP is known, the packet is flooded in a dense mode fashion. If the multicast traffic from a specific source is sufficient, the first hop router of the receiver may send Join messages toward the source to build a source-based distribution tree.

PIM-SM distributes information about active sources by forwarding data packets on the shared tree. Because PIM-SM uses shared trees (at least, initially), it requires the use of a rendezvous point (RP). The RP must be administratively configured in the network.

In sparse mode, a router assumes that other routers do not want to forward multicast packets for a group, unless there is an explicit request for the traffic. When hosts join a multicast group, the directly connected routers send PIM Join messages toward the RP. The RP tracks multicast groups. Hosts that send multicast packets are registered with the RP by the first hop router of that host. The RP then sends Join messages toward the source. At this point, packets are forwarded on a shared distribution tree. If the multicast traffic from a

specific source is sufficient, the first hop router of the host may send Join messages toward the source to build a source-based distribution tree.

First-hop designated routers with directly connected sources register with the RP and then data is forwarded down the shared tree to the receivers. The edge routers learn about a particular source when they receive data packets on the shared tree from that source through the RP. The edge router then sends PIM (S, G) Join messages toward that source. Each router along the reverse path compares the unicast routing metric of the RP address to the metric of the source address. If the metric for the source address is better, it will forward a PIM (S, G) Join message toward the source. If the metric for the RP is the same or better, then the PIM (S, G) Join message will be sent in the same direction as the RP. In this case, the shared tree and the source tree would be considered congruent.

If the shared tree is not an optimal path between the source and the receiver, the routers dynamically create a source tree and stop traffic from flowing down the shared tree. This behavior is the default behavior in Cisco IOS software. Network administrators can force traffic to stay on the shared tree by using the Cisco IOS **ip PIM spt-threshold infinity** command.

PIM-SM scales well to a network of any size, including those with WAN links. The explicit join mechanism prevents unwanted traffic from flooding the WAN links.

Vif1 interface

A Vif1 interface is a logical IP interface that

- remains always up when the router is active, similar to a loopback interface
- resides on its own unique subnet that should be advertised in Interior Gateway Protocol (IGP) updates (RIP, EIGRP, OSPF, ISIS), and
- maintains information about the input interface, private-to-public mgroup mappings, mask length, which defines your pool range, and the source of the translated packet.

Multicast service reflection application

A multicast service reflection application is a Cisco IOS software application that

- runs in Cisco IOS software interrupt level switching and processes packets forwarded to the Vif1 interface
- translates both source and destination addresses, unlike IP multicast Network Address Translation (NAT) which only translates the source IP address, and
- enables users who have not yet moved to the new multicast group to receive the untranslated stream.

Implementation and operation

Multicast service reflection is implemented using an interface CLI statement. Each configured multicast service reflection CLI statement establishes a packet match and rewrite operation acting on packets sent by Cisco IOS unicast or multicast packet routing onto the Vif1 interface. The matched and rewritten packet is sent back into Cisco IOS unicast or multicast packet routing, where it is handled like any other packet arriving from an interface.

The Vif1 interface is a receiver for the original stream and makes it appear that the new stream is coming from a source directly connected to the Vif1 subnet. The Vif1 interface is a Designated Router (DR) for active sources and registers with the appropriate RP.

More than one multicast service reflection operation can be configured to match the same packets, which allows you to replicate the same received traffic to multiple destination addresses.

How to implement multicast service reflection

This reference provides the necessary configuration steps and procedures to implement multicast service reflection.

Configure multicast service reflection

Use this task to configure multicast service reflection to match and rewrite multicast packets on interfaces, enabling packet matching and rewriting on multicast interfaces.

Procedure

Step 1 Use the **ip multicast-routing** *[distributed]* command to enable IP multicast routing.

Use the **distributed** keyword to enable the Multicast Distributed Switching feature.

Example:

```
Router(config)# ip multicast-routing
```

Step 2 Use the **interface** *type number* command to enter interface configuration mode for the specified interface type and number.

Example:

```
Router(config)# interface ethernet 0
```

Step 3 Use the **ip pim sparse-mode** command to enable PIM sparse mode on the interface.

Example:

```
Router(config-if)# ip pim sparse-mode
```

Step 4 Use the **no shutdown** command to enable the interface.

Example:

```
Router(config-if)# no shutdown
```

a) Use the **exit** command to exit interface configuration mode and return to global configuration mode.

Example:

```
Router(config-if)# exit
```

Repeat Step 4 through Step 7 for each PIM interface.

Step 5 Use the **interface Vif1** command to enter interface configuration mode for the Vif1 interface.

Example:

```
Router(config)# interface Vif1
```

Step 6 Use the **ip address** *ip-address mask [secondary]* command to set a primary or secondary IP address for the interface.

Example:

```
Router(config-if)# ip address 10.1.1.1 255.255.255.0
```

Step 7 Use the **ip pim sparse-mode** command to enable PIM sparse mode on the interface.

Example:

```
Router(config-if)# ip pim sparse-mode
```

Step 8 Use the **ip service reflect** *input-interface* **destination** *destination-address* **to** *new-destination-address* **mask-len** *number* **source** *new-source-address* command to match and rewrite multicast packets routed onto the Vif1 interface.

The matched and rewritten packets are sent back into Cisco multicast packet routing (or unicast routing if the destination is unicast), where they are handled like any other packets arriving from an interface.

Example:

```
Router(config-if)# ip service reflect ethernet0 destination 224.1.1.0 to 239.2.2.0 mask-len
 24 source 10.1.1.2
```

Step 9 Use the **ip igmp static-group** *{* | *group-address* **source** *{source-address | ssm-map}}* command to configure the router as a statically connected member of the specified group on the interface, and to forward traffic destined for the multicast group onto the interface.

This step is applicable only for multicast-to-multicast and multicast-to-unicast scenarios.

Example:

```
Router(config-if)# ip igmp static-group 224.1.1.1
```

Multicast service reflection is configured, and the router can match and rewrite multicast packets on the specified interfaces.

Configuration examples for multicast service reflection

This topic provides configuration examples for implementing multicast service reflection to assist in network setup.

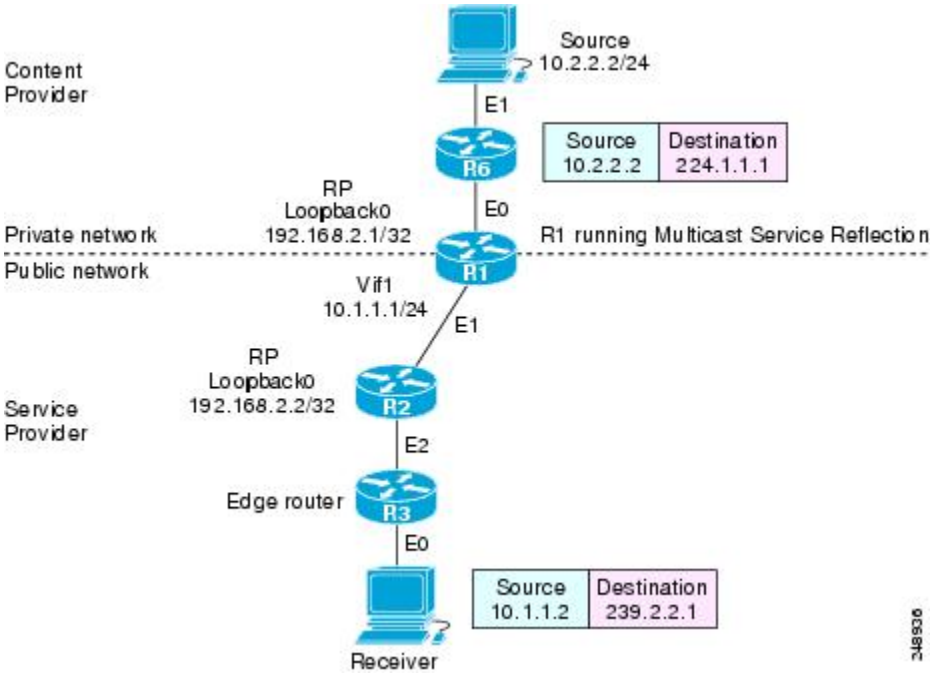
See the following examples.

Example multicast-to-multicast destination translation

This example shows how to implement multicast service reflection (multicast-to-multicast destination translation) in a service provider network. Multicast-to-Multicast Destination Translation allows service providers to translate externally received content provider multicast destination addresses to multicast destination addresses that conform to the service provider's internal addressing policy.

This example uses the topology illustrated in the figure.

Figure 3: Multicast service reflection (Multicast-to-Multicast destination translation) in a service provider network example topology



In this example topology, a content provider is sending financial market information to a service provider, which in turn is sending that information to active receivers (brokerage houses). The service provider may be receiving market data from multiple content providers.

Router R1 is an edge router in the service provider's PIM domain.

Router R1 has a loopback interface and is acting as the RP for the 224.1.1.0/24 address range.

Router R1 has a Vif1 interface and is running the multicast service reflection application.

Router R2 has a loopback interface and is acting as the RP for the 239.2.2.0/24 address range.

Enter these commands on the router running the multicast service reflection application (R1):

```
!
configure terminal
ip multicast-routing
interface &lt;all IP numbered interfaces&gt;
ip pim sparse-mode
no shutdown
!
! Configure the loopback interface for the Service Provider RP
!
interface loopback 0
ip address 192.168.2.1 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.2.1 mcast-content-provider-groups override
ip pim rp-address 192.168.2.2 mcast-service-provider-groups override
ip access-list standard mcast-content-provider-groups permit 224.1.1.0 0.0.0.255
ip access-list standard mcast-service-provider-groups permit 239.2.2.0 0.0.0.255
!
! Configure the Vif1 virtual interface for multicast service reflection
!
interface Vif1
```

```
ip address 10.1.1.1 255.255.255.0
ip pim sparse-mode
ip service reflect Ethernet 0 destination 224.1.1.0 to 239.2.2.0 mask-len 24 source 10.1.1.2
ip igmp static-group 224.1.1.0
ip igmp static-group 224.1.1.1
ip igmp static-group 224.1.1.2
ip igmp static-group 224.1.1.3
.
.
.
ip igmp static-group 224.1.1.255
```

Enter these commands on the router that is the RP in the service provider network (R2):

```
ip multicast-routing
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
interface loopback 0
ip address 192.168.2.2 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.2.2 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups permit 239.2.2.0 0.0.0.255
!
```

Enter these commands on all the other routers in the service provider network (R3):

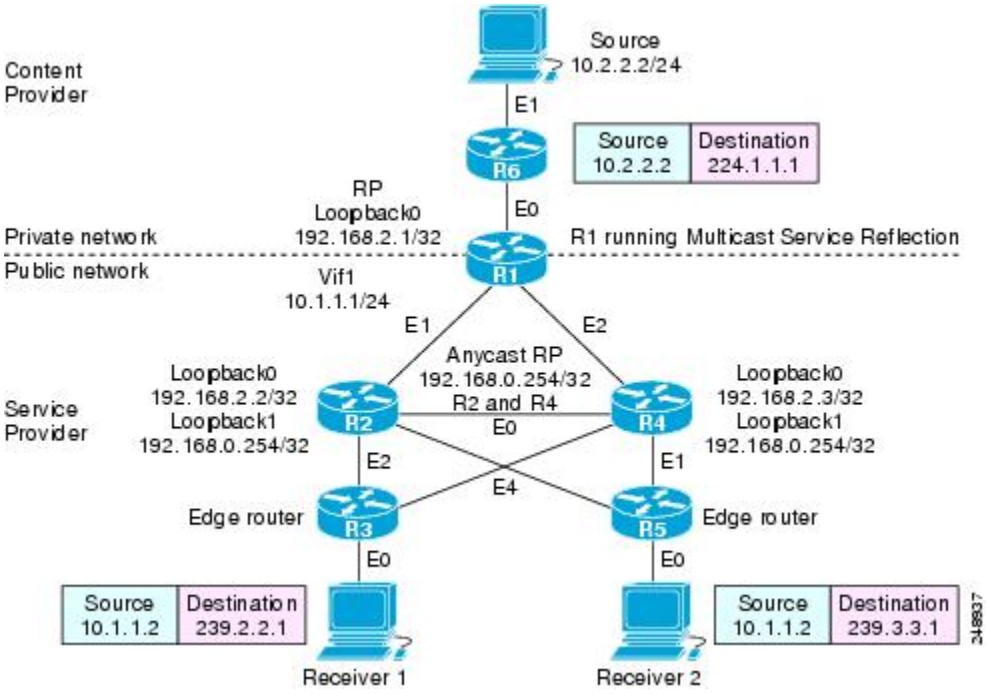
```
ip multicast-routing
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!
ip pim rp-address 192.168.2.2 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups permit 239.2.2.0 0.0.0.255
end
```

Example multicast-to-unicast destination translation

This example shows how to implement multicast service reflection (multicast-to-unicast destination translation) in a service provider network. Multicast-to-Unicast Destination Translation allows service providers to translate externally received content provider multicast destination addresses to unicast destination addresses that conform to the service provider's internal addressing policy.

This example uses the topology illustrated in the figure.

Figure 4: Multicast service reflection (Multicast-to-Unicast destination translation) in a service provider network example topology



In this example topology, a content provider is sending financial market information to a service provider, which in turn is sending that information to active receivers (brokerage houses). The service provider may be receiving market data from multiple content providers.

Router R1 is an edge router in the service provider's PIM domain.

Router R1 has a loopback interface and is acting as the RP for the 224.1.1.0/24 address range.

Router R1 has a Vif1 interface and is running the multicast service reflection application.

Routers R2 and R3 are non PIM enabled routers running unicast routing only in the service provider network.

Enter these commands on the router running the multicast service reflection application (R1):

```
configure terminal
ip multicast-routing
interface <all IP numbered interfaces>
ip pim sparse-mode
no shutdown
!

! Configure the loopback interface for the Service Provider RP
!
interface loopback 0
ip address 192.168.2.1 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.2.1 mcast-content-provider-groups override
ip access-list standard mcast-content-provider-groups permit 224.1.1.10 0.0.0.255
!

! Configure the Vif1 virtual interface for multicast service reflection
```

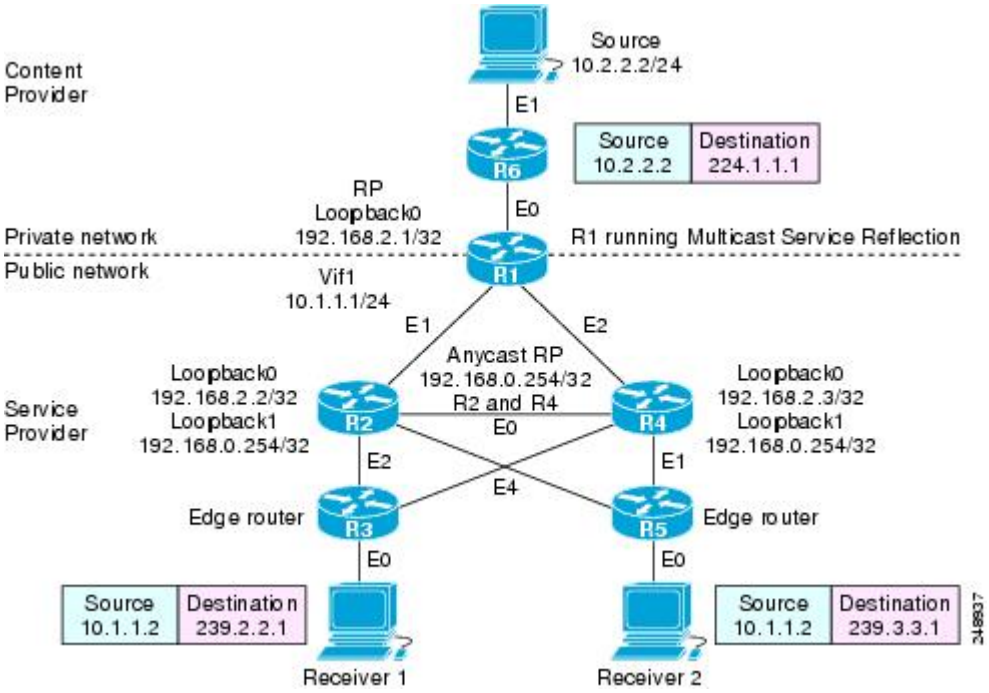
```
!
interface Vif1
ip address 10.1.1.1 255.255.255.0
ip pim sparse-mode
ip service reflect Ethernet 0 destination 224.1.1.0 to 10.3.3.0 mask-len 24 source 10.1.1.2
end
```

Configure multicast-to-multicast destination splitting

Implement multicast service reflection (multicast-to-multicast destination splitting) in a service provider network to translate externally received content provider multicast destination addresses to multiple multicast destination addresses that conform to the service provider's internal addressing policy.

This configuration allows service providers to translate externally received content provider multicast destination addresses to multiple multicast destination addresses that conform to the service provider's internal addressing policy. In this scenario, a content provider is sending financial market information to a service provider, which in turn is sending that information to active receivers (brokerage houses). The service provider may be receiving market data from multiple content providers.

Figure 5: Multicast service reflection (Multicast-to-Multicast destination splitting) in a service provider network example topology



- Router R1 is an edge router in the service provider's PIM domain
- Router R1 has a loopback configured and is acting as an RP for the 224.1.1.0/24 address range
- Router R1 has a Vif1 interface and is running the multicast service reflection application
- Routers R2 and R4 have multiple loopback interfaces and are acting as anycast RPs for the 239.2.2.0 and 239.3.3.0 address ranges
- Router R3 and R5 are edge routers in the service provider's PIM domain

Procedure

Step 1 Configure the router running the multicast service reflection application (R1).

Example:

```
configure terminal
ip multicast-routing
interface &lt;all IP numbered interfaces&gt;
ip pim sparse-mode
no shutdown
!
```

Step 2 Configure the loopback interface for the service provider RP on R1.

Example:

```
! Configure the loopback interface for the Service Provider RP
interface loopback 0
ip address 192.168.2.1 255.255.255.255
ip pim sparse-mode
ip pim rp-address 192.168.2.1 mcast-content-provider-groups override
ip pim rp-address 192.168.0.254 mcast-service-provider-groups override
ip access-list standard mcast-content-provider-groups permit 224.1.1.0 0.0.0.255
ip access-list standard mcast-service-provider-groups permit 239.2.2.0 0.0.0.255
ip access-list standard mcast-service-provider-groups permit 239.3.3.0 0.0.0.255
!
```

Step 3 Configure the Vif1 virtual interface for multicast service reflection on R1.

Example:

```
! Configure the Vif1 virtual interface for multicast service reflection
interface Vif1
ip address 10.1.1.1 255.255.255.0
ip pim sparse-mode
ip service reflect Ethernet 0 destination 224.1.1.0 to 239.2.2.0 mask-len 24 source 10.1.1.2
ip service reflect Ethernet 0 destination 224.1.1.0 to 239.3.3.0 mask-len 24 source 10.1.1.2
ip igmp static-group 224.1.1.0
ip igmp static-group 224.1.1.1
ip igmp static-group 224.1.1.2
ip igmp static-group 224.1.1.3
ip igmp static-group 224.1.1.254
```

Step 4 Configure the R2 router that is an anycast RP in the service provider network.

Example:

```
ip multicast-routing
interface &lt;all IP numbered interfaces&gt;
ip pim sparse-mode
no shutdown
interface loopback 0
ip address 192.168.2.2 255.255.255.255
ip pim sparse-mode
interface loopback 1
description --- Anycast RP ---
ip address 192.168.0.254 255.255.255.255
ip pim sparse-mode
ip msdp peer 192.168.2.3 connect-source Loopback0
ip msdp originator-id Loopback0
ip pim rp-address 192.168.0.254 mcast-service-provider-groups override
```

```
ip access-list standard mcast-service-provider-groups permit 239.2.2.0 0.0.0.255
ip access-list standard mcast-service-provider-groups permit 239.3.3.0 0.0.0.255
```

Step 5 Configure the R4 router that is an anycast RP in the service provider network.

Example:

```
ip multicast-routing
interface &lt;all IP numbered interfaces&gt;
ip pim sparse-mode
no shutdown
!
interface loopback 0
ip address 192.168.2.3 255.255.255.255
ip pim sparse-mode
interface loopback 1
ip address 192.168.0.254 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.0.254 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups permit 239.2.2.0 0.0.0.255
ip access-list standard mcast-service-provider-groups permit 239.3.3.0 0.0.0.255
!
ip msdp peer 192.168.2.2 connect-source Loopback0
ip msdp originator-id Loopback0
```

Step 6 Configure the R3 and R5 routers in the service provider network.

Example:

```
ip multicast-routing
ip pim rp-address 192.168.0.254 mcast-service-provider-groups override
ip access-list standard mcast-service-provider-groups permit 239.2.2.0 0.0.0.255
permit 239.3.3.0 0.0.0.255
!
```

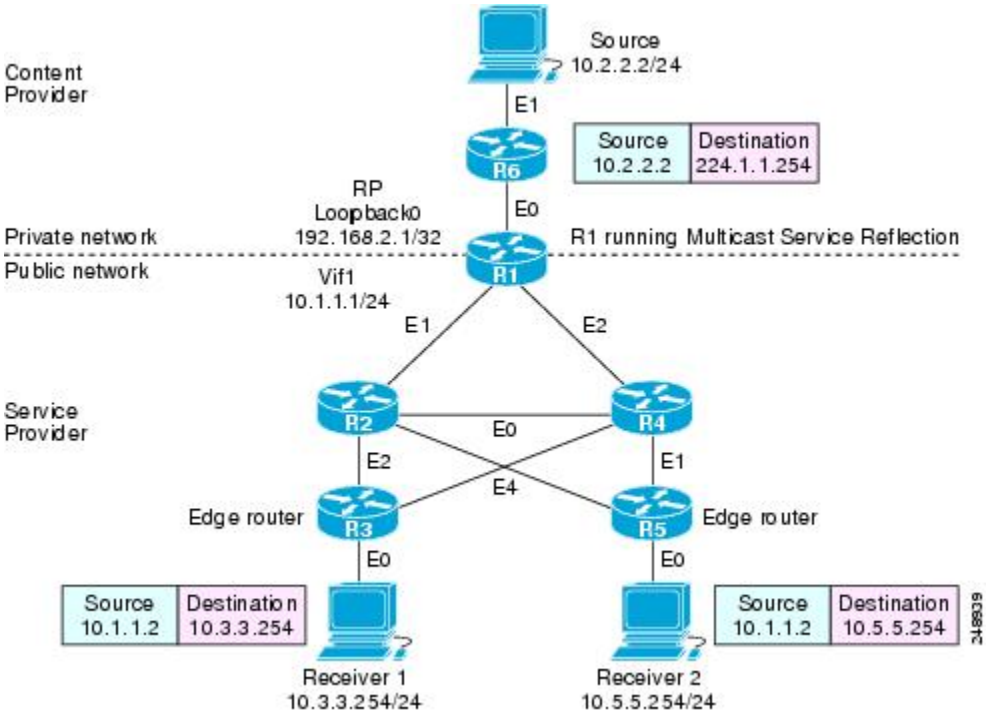
The multicast service reflection configuration is complete. The service provider can now translate externally received content provider multicast destination addresses to multiple multicast destination addresses that conform to the internal addressing policy, allowing the single multicast stream to be converted into two unique multicast streams.

Example multicast-to-unicast destination splitting

This example shows how to implement multicast service reflection (multicast-to-unicast destination splitting where the multicast single stream is converted into two unique unicast streams) in a service provider network. Multicast-to-Unicast Destination Splitting allows service providers to translate externally received content provider multicast destination addresses to multiple unicast destination addresses that conform to the service provider's internal addressing policy.

This example uses the topology illustrated in this figure.

Figure 6: Multicast service reflection (Multicast-to-Unicast destination splitting) in a service provider network example topology



In this example topology, a content provider is sending financial market information to a service provider, which in turn is sending that information to active receivers (brokerage houses). The service provider may be receiving market data from multiple content providers.

Router R1 is an edge router in the service provider's PIM domain.

Router R1 is acting as a RP for the 224.1.1.0/24 address range.

Router R1 has a Vif1 interface and is running the multicast service reflection application.

Routers R2, R3, R4 and R5 are not PIM enabled and are running unicast routing only in the service provider network.

Enter these commands on the router running the multicast service reflection application (R1):

```
configure terminal
ip multicast-routing
interface &lt;all IP numbered interfaces&gt;
ip pim sparse-mode
no shutdown
!
! Configure the loopback interface for the Service Provider RP
!
interface loopback 0
ip address 192.168.2.1 255.255.255.255
ip pim sparse-mode
!
ip pim rp-address 192.168.2.1 mcast-content-provider-groups override
ip access-list standard mcast-content-provider-groups permit 224.1.1.0 0.0.0.255
!
! Configure the Vif1 virtual interface for multicast service reflection
!
interface Vif1
```

```
ip address 10.1.1.1 255.255.255.0
ip pim sparse-mode
ip service reflect Ethernet 0 destination 224.1.1.0 to 10.3.3.0 mask-len 24 source 10.1.1.2
ip service reflect Ethernet 0 destination 224.1.1.0 to 10.5.5.0 mask-len 24 source 10.1.1.2
ip igmp static-group 224.1.1.0
ip igmp static-group 224.1.1.1
ip igmp static-group 224.1.1.2
ip igmp static-group 224.1.1.3
.
.
.
ip igmp static-group 224.1.1.255
!
end
```

Configure PIMv6 Anycast RP solution

Use this task to configure PIMv6 anycast RP1 and RP2 peers.

Procedure

Step 1 Configure RP1 using the given steps.

a) Use the **ipv6 address** [*ipv6-address/prefix-length* | *prefix-name sub-bits /prefix-length*] to configure an IPv6 address based on an IPv6 general prefix and to enable IPv6 processing on an interface.

Example:

```
Router(config-if)# ipv6 add 2001:DB8::1:1/128
```

b) Use the **interface** [*type*] [*number*] to specify the interface type and number.

Example:

```
Router(config-if)# interface Loopback4
```

c) Use the **ipv6 address** [*ipv6-address/prefix-length* | *prefix-name sub-bits /prefix-length*] to configure an IPv6 address based on an IPv6 general prefix and to enable IPv6 processing on an interface.

Example:

```
Router(config-if)# ipv6 address 2001:DB8::4:4/64
```

d) Use the **no shut** to enable the interface.

Example:

```
Router(config-if)#  no shut
```

e) Use the **interface** [*type*] [*number*] to specify the interface type and number.

Example:

```
Router(config-if)# interface Loopback5
```

f) Use the **ipv6 address vrf** [*vrf -name*] **rp-address** [*ipv6-address*] [*group-address-list*] [**bidir**] to configure the address of a PIM RP for a particular group range.

Example:

```
Router(config-if)# ipv6 pim rp-address 2001:DB8::1:1 acl_sparse1
```

g) Use the **no shut** to enable the interface.

Example:

```
Router(config-if)# no shut
```

h) Use the **exit** command to exit interface configuration mode and enter the global configuration mode.

Example:

```
Router(config-if)# exit
```

Step 2 Use these commands to configure the address of the PIM RP:

a) Use the **ipv6 pim anycast-RP** [*rp-address*] [*peer-address*] to configure the address of the PIM RP for an anycast group range.

Example:

```
Router(config-if)# ipv6 pim anycast-rp 2001:DB8::1:1 2001:DB8::3:3
```

b) Use the **ipv6 pim anycast-RP** [*rp-address*] [*local-address*] to configure the address of the PIM RP for an anycast group range.

Example:

```
Router(config-if)# ipv6 pim anycast-rp 2001:DB8::1:1 2001:DB8::4:4
```

Step 3 Configure RP2 using the given steps.

a) Use the **ipv6 address** [*ipv6-address/prefix-length* | *prefix-name sub-bits /prefix-length*] to configure an IPv6 address based on an IPv6 general prefix and to enable IPv6 processing on an interface.

Example:

```
Router(config-if)# ipv6 add 2001:DB8::1:1/128
```

b) Use the **interface** [*type*] [*number*] to specify the interface type and number.

Example:

```
Router(config-if)# interface Loopback4
```

c) Use the **ipv6 address** [*ipv6-address/prefix-length* | *prefix-name sub-bits /prefix-length*] to configure an IPv6 address based on an IPv6 general prefix and to enable IPv6 processing on an interface.

Example:

```
Router(config-if)# ipv6 address 2001:DB8::3:3/64
```

d) Use the **no shut** to enable the interface.

Example:

```
Router(config-if)# no shut
```

e) Use the **interface** [*type*] [*number*] to specify the interface type and number.

Example:

```
Router(config-if)# interface Loopback5
```

f) Use the **ipv6 address vrf** [*vrf -name*] **rp-address** [*ipv6-address*] [*group-address-list*] [**bidir**] to configure the address of a PIM RP for a particular group range.

Example:

```
Router(config-if)# ipv6 pim rp-address 2001:DB8::1:1 acl_sparse1
```

g) Use the **no shut** to enable the interface.

Example:

```
Router(config-if)# no shut
```

h) Use the **ipv6 pim anycast-RP** [*rp-address*] [*peer-address*] to configure the address of the PIM RP for an anycast group range.

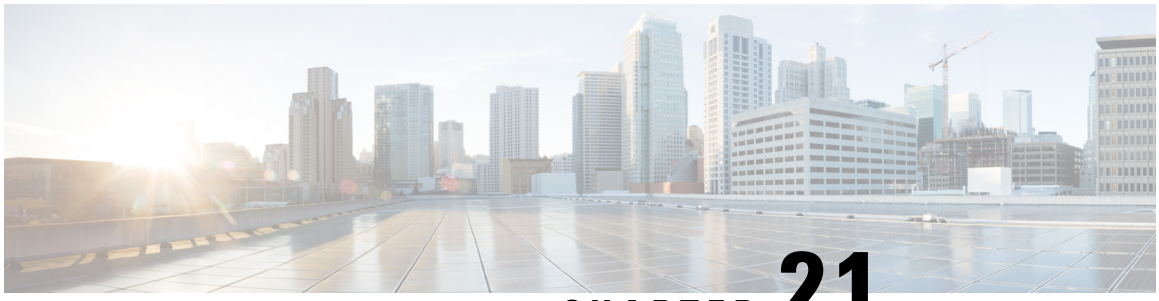
Example:

```
Router(config-if)# ipv6 pim anycast-rp 2001:DB8::1:1 2001:DB8::3:3
```

i) Use the **ipv6 pim anycast-RP** [*rp-address*] [*local-address*] to configure the address of the PIM RP for an anycast group range.

Example:

```
Router(config-if)# ipv6 pim anycast-rp 2001:DB8::1:1 2001:DB8::4:4
```



CHAPTER 21

Cellular Boot Time Improvements

• Cellular boot time improvements, on page 189

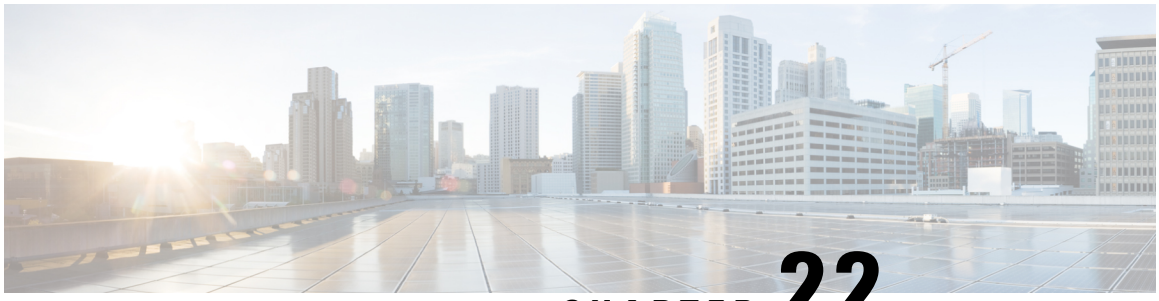
Cellular boot time improvements

Cellular boot time improvements are performance enhancements that

- reduce the time required for cellular interfaces to become operational after router startup
- decrease cellular link up-time by approximately 20% compared to previous releases, and
- enable faster traffic passing capabilities following system initialization.

Performance improvements

In previous releases, the cellular interface was taking approximately two and a half minutes to come up and pass traffic after the router booted up. The cellular link up-time has been improved by approximately 20% in IOS-XE release 17.9.1.



CHAPTER 22

Cellular Pluggable Interface Modules

This document provides reference information regarding the deployment and technical characteristics of cellular pluggable interface modules.

- Cellular pluggable interface modules, on page 191
- Support for P-5GS6-R16SA-GL pluggable module, on page 191
- Support for the P-5GS6-GL pluggable module, on page 192
- Galileo support on the LTE pluggable modules, on page 192
- Radio signal parameters and GPS coordinates of cellular telemetry monitoring using syslog messages, on page 193
- GPS NMEA unique identifier , on page 195
- Configure cellular modem GPS NMEA unique identifier, on page 196

Cellular pluggable interface modules

The Cisco 4G LTE-Advanced Configuration chapter has been replaced by a new standalone guide called Cellular Pluggable Interface Module Configuration Guide . This guide contains updated information on all aspects of using the Cisco Cellular PIM.

Important The Pluggable Module is not hot swappable. The router must be reloaded after a new module is installed.

Support for P-5GS6-R16SA-GL pluggable module

From Cisco IOS-XE Release 17.18.2, the P-5GS6-R16SA-GL Pluggable Module is supported on ESR6300 routers.

Support for the P-5GS6-R16SA-GL Pluggable Module works the same way on the ESR6300 routers as it does on the other IoT routers. For details, see 5G Sub-6 GHz Pluggable Interface Module and Cellular Pluggable Interface Module Configuration Guide .

Support for the P-5GS6-GL pluggable module

Support for the P-5GS6-GL Pluggable Module works the same on the ESR6300 as it does on the other IoT Routers.

For details, see 5G Sub-6 GHz Pluggable Interface Module and Cellular Pluggable Interface Module Configuration Guide .

Galileo support on the LTE pluggable modules

The Cisco IOS XE 17.11.1a release introduces support for the Galileo constellation on LTE pluggable modules.

GPS was the only GNSS constellation supported in earlier releases. The Cisco IOS XE 17.11.1a release introduces support for Galileo constellation.



Note You can enable only one constellation at a time.

Configuration command examples:

```
config# controller cellular<slot/port>
(config-controller)# <no> lte gps constellation <gps | galileo | gnss >
```

Example:

```
(config-controller)# lte gps constellation ?
galileo  select Galileo as active constellation
gps      select GPS as active constellation
gnss      select multiple GNSS as active constellation
```



Note The default setting is GPS mode.

The galileo and gnss options in this CLI configure Galileo and simultaneous GNSS modes, such as GPS and Galileo.

If you disable the GPS configuration, make sure that no constellation is configured. This maintains consistency with GPS mode configuration.

Example:

```
config#
controller Cellular 0/3/0
```

```
(config-controller)# no lte gps constellation gps
```

Show commands example:

Note The example shows the current GNSS constellation as Galileo.

Example:

```
config#
show cellular 0/3/0 gps detail



GPS Feature =  enabled
GPS Mode Configured =  standalone
Current Constellation Configured =  galileo
GPS Port Selected =  Dedicated GPS port
GPS Status =  GPS acquiring
```

Any changes made to the configuration will require the router to be rebooted.

More information is available in the Cellular Pluggable Interface Module Configuration Guide .

Radio signal parameters and GPS coordinates of cellular telemetry monitoring using syslog messages

Cellular telemetry is a feature that enables real-time monitoring and analysis of cellular connection performance by collecting Radio Frequency (RF) parameters and Global Positioning System (GPS) coordinates from the cellular network at a fixed interval of 60 seconds.

- Monitors key radio signal parameters such as RSSI, RSRP, RSRQ, PCI, and SNR.
- Collects GPS coordinates for location tracking.
- Facilitates troubleshooting, network optimization, and ensures reliable connectivity.

Cellular Telemetry Reference Information

Cellular telemetry parameters that can be monitored include:

- Received Signal Strength Indicator (RSSI)
- Reference Signal Received Power (RSRP)
- Reference Signal Received Quality (RSRQ)
- Physical Cell Identity (PCI)
- Signal to Noise Ratio (SNR)
- Global Positioning System (GPS) coordinates

The Cisco Catalyst WAN Manager does not support cellular telemetry.

Enable cellular telemetry

To enable the cellular telemetry feature in the contoller cellular interface 0/x/0 using the CLI.

Before you begin

- Insert the cellular Pluggable Interface Module (PIM) into the IR device.
- Enable GPS in the controller for GPS coordinates to be displayed in syslogs.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Router#configure terminal
```

Step 2 Enter the cellular configuration mode.

Example:

```
Router (config)#controller cellular 0/3/0
```

Step 3 Enable the RF parameters and GPS coordinates.

Example:

```
Router (config-controller)#lte modem serviceability signal-parameters
Router (config-controller)#end
```

Disable cellular telemetry

To disable the cellular telemetry feature, use the no form of the **lte modem serviceability signal-parameters** command, as shown in the example:

```
Router#Configure terminal

Router(config)#controller cellular 0/3/0


Router(config-controller)#no lte modem serviceability signal-parameters
Router(config-controller)#end
```

Monitor cellular telemetry

You can monitor RF parameters and GPS coordinates for cellular interfaces by using the **show logging** command or check the console output.

The following example displays RF parameters and GPS coordinates at every one minute interval:

```
Router#show logging
*Sep  3 17:08:42.081: %CELLWAN-2-MODEM_SIGNAL_PARAM: Cellular0/3/0: 4G: RSSI = -54 dBm RSRP
```

```
 = -76 dBm RSRQ = -9 dB PCI = 1 SNR = 27.4 dB
Latitude =  12 Deg 56 Min 8.9260 Sec North
Longitude =  77 Deg 41 Min 44.1641 Sec East

*Sep  3 17:09:42.080: %CELLWAN-2-MODEM_SIGNAL_PARAM: Cellular0/3/0: 4G: RSSI = -54 dBm RSRP
 = -76 dBm RSRQ = -9 dB PCI = 1 SNR = 29.0 dB
Latitude =  12 Deg 56 Min 8.8989 Sec North
Longitude =  77 Deg 41 Min 44.1570 Sec East
```

The following example displays only RF parameters and not GPS coordinates since GPS is not enabled in the controller:

```
Router#show logging
*Sep  3 17:08:42.081: %CELLWAN-2-MODEM_SIGNAL_PARAM: Cellular0/3/0: 4G: RSSI = -54 dBm RSRP
 = -76 dBm RSRQ = -9 dB PCI = 1 SNR = 27.4 dB

*Sep  3 17:09:42.080: %CELLWAN-2-MODEM_SIGNAL_PARAM: Cellular0/3/0: 4G: RSSI = -54 dBm RSRP
 = -76 dBm RSRQ = -9 dB PCI = 1 SNR = 29.0 dB
```

GPS NMEA unique identifier

Starting with Cisco IOS-XE Release 26.1.1, Cisco routers can send a Unique Identifier (UID) to identify the source device when streaming NMEA GPS sentences over UDP on the Cisco IR1101 and Cisco IR1835 routers. UID streaming is supported from the Dead Reckoning (DR) module and from cellular modems that support GNSS. The UID uniquely identifies each router in GPS data processing systems.

Table 19: Feature History Table

Feature name	Release information	Feature description
Send a UID to identify the unique source when sending GPS coordinates data	Release 26.1.1	This feature allows you to configure Cisco routers to send a Unique Identifier (UID) while streaming NMEA GPS sentences over UDP. This UID streaming is supported by both the Dead Reckoning (DR) module and cellular modems with GNSS capabilities, enabling each router to be uniquely identified in GPS data processing systems.

Cisco proprietary NMEA sentence format

When you configure a UID, a Cisco proprietary NMEA sentence is sent alongside the standard NMEA sentence in the same UDP packet. The format is:

```
$PCSCU,<UID>,<Source>,<Router Time>,<NMEA Sentence Time>,<NMEA Sentence Checksum>,,,,,,*95
```

- `CSC` indicates Cisco
- `U` indicates this sentence provides a unique identifier
- `<UID>` is the serial number, hostname, or custom ID
- `<Source>` indicates the origin of the NMEA stream:

- `0`: cellular modem0
- `1`: cellular modem1
- `2`: DR module

For example, for cellular modem1, if you use the serial number as the UID, the Cisco proprietary NMEA sentence appears as:

```
$PCSCU,FCW2447P0EU,1,213431.100,213431.000,75,,,,,,*95
```

In this NMEA sentence

- `CSC` indicates Cisco
- `U` indicates that this sentence provides a unique identifier
- `FCW2447P0EU` indicates serial number as the UID
- `1` indicates source as cellular modem1
- `1213431.100` indicates the router time
- `213431.000` indicates the NMEA sentence time
- `75` indicates the NMEA sentence checksum

NMEA sentence selection

By default, all NMEA sentences from the modem or DR module are sent to the configured UDP destination. To reduce UDP traffic, you can select specific NMEA sentence types to send using a hex bitmask. The table presents each NMEA sentence type and its corresponding HEX value.

Table 20: NMEA sentence type and HEX value

HEX value	NMEA sentence type
0x01	GGA
0x02	GSA
0x04	GSV
0x08	RMC
0x10	VTG
0x20	GNS

Configure cellular modem GPS NMEA unique identifier

Follow these steps to configure cellular modem GPS NMEA UID.

Procedure

Step 1 Use the **configure terminal** command to enter the configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **controller cellular** *slot* command to enter the controller cellular configuration mode.

Example:

```
Router(config)# controller cellular 0/x/0
```

Step 3 Use the **lte gps nmea uid** {**custom** *unique-identifier* **| hostname | serial-number**} command to enable the GPS NMEA UID.

UID can be any of these:

- **custom**: user-configured string of upto 50 characters.
- **hostname**: host name of the router.
- **serial-number**: serial number of the router.

When using **serial-number** or **hostname**, the UID is automatically extracted and added to the proprietary NMEA sentence. This allows uniform CLI configuration across routers.

When using a **custom ID**, each router must be configured with a *unique-identifier* value to ensure distinct identification.

Example:

```
Router(config-controller)# lte gps nmea uid custom 1428
```

Example:

```
Router(config-controller)# lte gps nmea uid hostname
```

Example:

```
Router(config-controller)# lte gps nmea uid serial-number
```

Step 4 Use the **lte gps nmea ip udp** *source-address destination-address destination-port* **stream** *stream-id* command to specify the source and destination IP address.

Example:

```
Router(config-controller)# lte gps nmea ip udp 10.195.79.179 171.70.55.77 14013 stream 1
```

Step 5 Use the **lte gps nmea filter** *hex-value* command to select the sentence types.

hex-value: HEX bitmask for NMEA message filtering (0x01:GGA, 0x02:GSA, 0x04:GSV, 0x08:RMC, 0x010:VTG, 0x20:GNS)

This example shows the HEX value if you want to send sentence type as only RMC (0x08) and GNS (0x20):

Example:

```
Router(config-controller)# lte gps nmea filter 28
```

This example shows the HEX value if you want to send sentence type as only GGA (0x01) and GSA (0x02):

Example:

```
Router(config-controller)# lte gps nmea filter 03
```

Step 6 Use the **end** command to exit the controller configuration mode and return to the privileged EXEC mode.

Example:

```
Router(config-controller)# end
```



CHAPTER 23

Radio Aware Routing and Dynamic Link Exchange Protocol

This chapter contains the following sections:

- Radio aware routing, on page 200
- Benefits of radio aware routing, on page 202
- Restrictions for radio aware routing and DLEP features, on page 202
- Performance, on page 202
- System components, on page 203
- Radio aware routing topology, on page 204
- QoS provisioning on PPPoE extension session, on page 207
- Configure the RAR feature in bypass mode, on page 207
- Verifying RAR Session Details, on page 209
- DLEP, on page 214
- Configuring DLEP with OSPFv3, on page 216
- DLEP configuration modes, on page 220
- DLEP configuration with GTSM, on page 227
- Configure DLEP with OSPFv3, on page 228
- Configure DLEP with EIGRP, on page 230
- DLEP with quality of service (QoS), on page 231
- Configure DLEP on a sub-interface, on page 236
- Remove the DLEP configuration, on page 239
- Configure DLEP using the WebUI, on page 240
- DLEP validation commands, on page 241
- Optional configurations for DLEP, on page 244
- Restrictions for DLEP IPv6 unicast, on page 244
- IPv4 and IPv6 multicast over DLEP, on page 245
- IPv6 control plane for DLEP, on page 246
- DLEP client and neighbor clearing, on page 247
- Troubleshoot with show commands, on page 248
- Troubleshoot with debug commands, on page 250
- Additional debug commands, on page 252
- Related documentation, on page 253

Radio aware routing

Radio aware routing is a mechanism that

- uses radios to interact with the routing protocol OSPFv3/EIGRP to signal the appearance, disappearance, and link conditions of one-hop routing neighbors,
- addresses the need for faster convergence, more efficient route selection, and better performance for delay-sensitive traffic in mobile ad hoc networks, and
- provides a solution for lengthy protocol timers that are not recommended in mobile networks.

Mobile ad hoc networking characteristics

Relief workers, soldiers, public safety personnel, and others need the right information, in the right place, at the right time, wherever they are located. Mobile ad hoc networks are emerging to address these needs. The IETF request for comment, RFC RFC5578, defines a PPP-over-Ethernet (PPPoE) based mechanism for integrating IP routers and mobile radios in ad hoc networks, enabling faster convergence, more efficient route selection, and better performance for delaysensitive traffic.

In large mobile networks, connections to the routing neighbors are often interrupted due to distance and radio obstructions. When these signals do not reach the routing protocols, protocol timers are used to update the status of a neighbor. Routing protocols have lengthy timers, which is not recommended in mobile networks.

The RAR feature is supported on the Cisco ESR6300 Embedded Series Router.

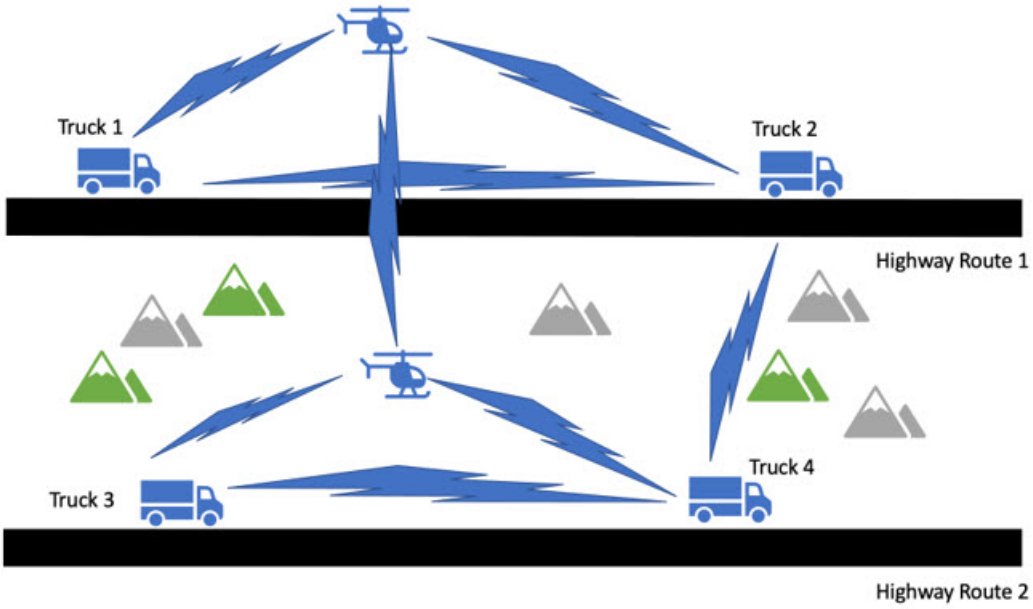
Mobile ad hoc networks are emerging as a means for delivering the benefits of IP networking to users operating beyond the reach of a fixed network. In ad hoc networks, mobile nodes associate on an extemporaneous or ad hoc basis. Ad hoc networks have numerous distinguishing characteristics when compared to conventional networking solutions:

- **Self-forming** — Nodes that come within radio range of each other can establish a network association without any pre-configuration or manual intervention.
- **Self-healing** — Nodes can join or leave rapidly without affecting operation of the remaining nodes.
- **No infrastructure** — In an ad hoc network, mobile nodes form their own network, and essentially become their own infrastructure.
- **Peer to peer** — Traditional networks typically support end systems operating in client-server mode. In an ad hoc network, mobile nodes can communicate and exchange information without prior arrangement and without reliance on centralized resources.
- **Predominantly wireless** — Historically networks have been mostly wired, and enhanced or extended through wireless access. The ad hoc environment is essentially wireless, but can be extended to support wired resources.
- **Highly dynamic** — Mobile nodes are in continuous motion and ad hoc networking topologies are constantly changing.

Collectively, these characteristics will enable ad hoc networks to deliver timely information to a new and underserved class of users. Ad hoc networking solutions can be applied to virtually any scenario that involves a cadre of highly mobile users or platforms (which may include stationary devices as well), a strong need to share IP-based information, and an environment in which fixed infrastructure is impractical, impaired, or impossible.

Real-world problem scenario

Consider the scenario in this network diagram:



The network is a voice, video, data network between moving vehicles that consists of both ground and air vehicles, hence the network is mobile and it is a peer to peer mesh that changes as topographical obstructions are encountered. This is where we get the term mobile ad hock network or MANET for short.

In the scenario in the drawing, all 4 trucks always have connectivity with the helicopters that are flying over the same road. The two helicopters always have line of sight and will always have a connection between each other. The trucks may even be able to connect to the other helicopter or a truck on the opposite road when conditions are favorable.

Here we see that the path between trucks 1 and 3 are completely blocked. The path between Truck 2 and 4 is about to be blocked.

Our existing routing protocols such as OSPFv3 and EIGRP need to adjust its path metrics very quickly to maintain a cohesive operational network. The routing protocol also needs a way to get that information from the radios and that requires a radio to router protocol that is delivered by Cisco Radio Aware Routing in the form of two open protocols:

- PPP over Ethernet (PPPoE)
- Dynamic Link Exchange Protocol (DLEP)

Both protocols are discussed later in this document.

Benefits of radio aware routing

The Radio Aware Routing feature offers multiple benefits for network performance and routing optimization.

- Provides faster network convergence through immediate recognition of changes.
- Enables routing for failing or fading radio links.
- Allows easy routing between line-of-sight and non-line-of-sight paths.
- Provides faster convergence and optimal route selection so that delay-sensitive traffic, such as voice and video, is not disrupted
- Provides efficient radio resources and bandwidth usage.
- Reduces impact on the radio links by performing congestion control in the router.
- Allows route selection based on radio power conservation.
- Enables decoupling of the routing and radio functionalities.
- Provides simple Ethernet connection to RFC 5578 and DLEP compliant radios.

Restrictions for radio aware routing and DLEP features

The Radio Aware Routing and DLEP features have these restrictions and limitations:

- Multicast traffic is not supported with DLEP, but is supported with PPPOE.
- DLEP does not support IPv6
- Cisco High Availability (HA) technology is not supported.
- Requires the Network Advantage license.



Note There is a condition that exists with DLEP that is not unique to the ESR6300. An ARP which has been learned on a VMI does not get deleted when the mapped physical interface goes down.



Note Previous releases of IOS XE offered support for IPv4 unicast traffic over an IPv4 DLEP session. IOS XE 17.12.1a provides support for IPv6 unicast, IPv4 multicast and IPv6 multicast over an IPv4 DLEP. See the Restrictions for DLEP IPv6 unicast, on page 244section.

Performance

The Radio Aware Routing feature supports a maximum of 10 neighbors per radio or VMI interface, with a total capacity of 30 to 40 neighbors.

System components

System components are infrastructure elements that

- implement the Radio Aware Routing (RAR) feature using the MANET (Mobile adhoc network) infrastructure
- include PPPOE, Virtual multipoint interface (VMI), QoS, routing protocol interface and RAR protocols, and
- provide communication mechanisms between radio and router components.

Component details

The system includes multiple specialized components:

- **Point-to-Point Protocol over Ethernet PPPOE or PPPOE**: PPPOE is a well-defined communication mechanism between the client and the server. In the RAR implementation, radio takes the role of the PPPOE client and router takes the role of the PPPOE server. This allows a loose coupling of radio and router, while providing a well-defined and predictable communication mechanism. As PPPOE is a session or a connection oriented protocol, it extends the point-to-point radio frequency (RF) link from an external radio to an IOS router.
- **PPPOE Extensions**: PPPOE extensions are used when the router communicates with the radio. In the Cisco IOS implementation of PPPOE, each individual session is represented by virtual access interface (connectivity to a radio neighbor) on which, QoS can be applied with these PPPOE extensions. RFC5578 provides extensions to PPPOE to support credit-based flow control and session-based real time link metrics, which are very useful for connections with variable bandwidth and limited buffering capabilities (such as radio links).
- **Virtual Multipoint Interface (VMI)**: Though PPPOE Extensions provides most of the setup to communicate between a router and a radio, VMI addresses the need to manage and translate events that higher layers (example, routing protocols) consume. In addition, VMI operates in the Bypass mode.

VMI supports two operational modes:

- **Bypass mode**: Every Virtual Access Interface (VAI) representing a radio neighbor is exposed to routing protocols OSPFv3 and EIGRP, so that, the routing protocol directly communicates with the respective VAI for both unicast and multicast routing protocol traffic.
- **Aggregate mode**: VMI is exposed to the routing protocols (OSPF) so that the routing protocols can leverage VMI for their optimum efficiency. When the network neighbors are viewed as a collection of networks on a point-to-multipoint link with broadcast and multicast capability at VMI, VMI helps in aggregating the multiple virtual access interfaces created from PPPOE. VMI presents a single multi access layer 2 broadcast capable interface. The VMI layer re-directs unicast routing protocol traffic to the appropriate P2P link (Virtual-Access interface), and replicates any Multicast/Broadcast traffic that needs to flow. Since the routing protocol communicates to a single interface, the size of the topology database is reduced, without impacting the integrity of the network.



Note DLEP only supports aggregate mode. The preferred mode for PPPOE is bypass.

Radio aware routing topology

A radio aware routing topology is a network architecture that

- integrates IP routing and mobile radio communications through device-to-radio links
- enables immediate notification when nodes join or leave the network, allowing devices to recognize topology changes faster than timer-based methods
- provides cross-layer feedback using PPPoE extensions defined in RFC 5578 for Ethernet-based communications between devices and mobile radios, and
- establishes PPPoE sessions on behalf of every device and radio neighbor discovered in the network to report radio network status.

Mobile ad hoc networks implementation

Mobile Ad Hoc Networks (MANETs) for device-to-radio communications address the challenges faced when merging IP routing and mobile radio communications in ad hoc networking applications.

Through the device-to-radio link, the radio can inform the device immediately when a node joins or leaves, and this enables the device to recognize topology changes more quickly than if it had to rely on timers. Without this link-status notification from the radio, the device would likely time out while waiting for traffic. The link-status notification from the radio enables the device to respond faster to network topology changes. Metric information regarding the quality of a link is passed between the device and radio, enabling the device to more intelligently decide on which link to use.

With the link-status signaling provided by the device-to-radio link, applications such as voice and video work better because outages caused by topology changes are reduced or eliminated. Sessions are more stable and remain active longer.

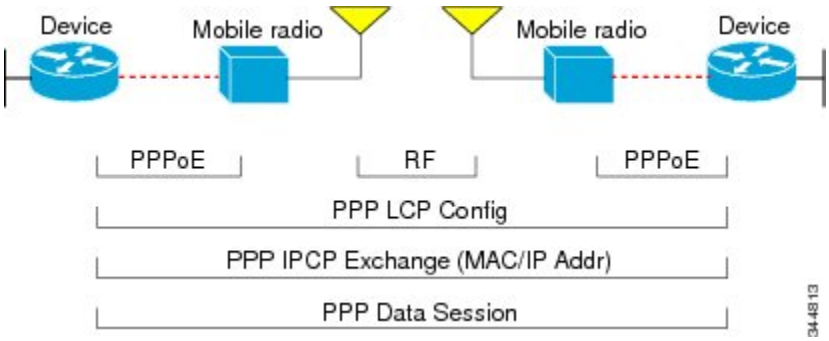
Cross-layer feedback for device-to-radio integration of Radio-Aware Routing (RAR) takes advantage of the functions defined in RFC 5578. The RFC 5578 is an Internet Engineering Task Force (IETF) standard that defines PPP over Ethernet (PPPoE) extensions for Ethernet-based communications between a device and a mobile radio, that operates in a variable-bandwidth environment and has limited buffering capabilities. These extensions provide a PPPoE session-based mechanism for sharing radio network status such as link-quality metrics and establishing flow control between a device and an RAR-compliant radio.

An RAR-compliant radio initiates a Layer 2 PPPoE session with its adjacent device on behalf of every device and radio neighbor discovered in the network. These Layer 2 sessions are the means by which radio network status for each neighbor link is reported to the device. The radio establishes the correspondence between each PPPoE session and each link to a neighbor.

PPPoE interfaces enable intranodal communications between devices and radios:

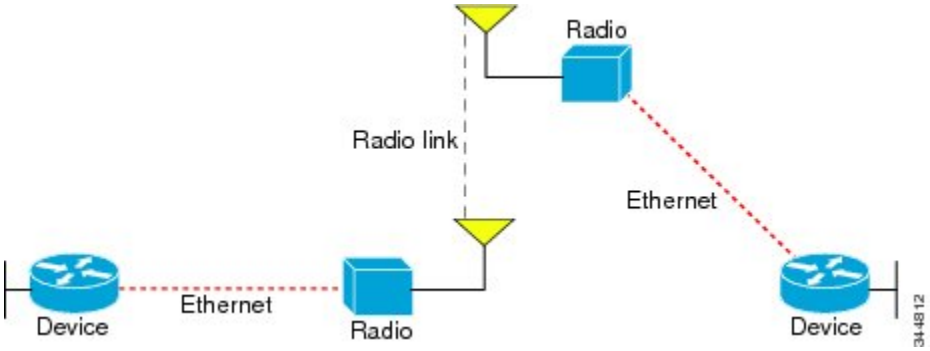
- Each radio initiates the PPPoE session as soon as the radio establishes a radio link to another radio
- After the PPPoE sessions are active, a PPP session is established end-to-end (device-to-device)
- The virtual multipoint interface (VMI) on the device can aggregate multiple PPPoE sessions and multiplex them to look like a single interface to the routing processes
- Virtual access interfaces are associated with each of the PPP and PPPoE connections underneath the VMI

Figure 7: PPPoE session exchange between mobile devices and directional radios



This capability requires that a Radio-Aware Routing (RAR)-compliant radio be connected to a device through Ethernet. The device always considers the Ethernet link to be up. If the radio side of the link goes down, the device waits until a routing update timeout occurs to declare the route down and then updates the routing table.

Figure 8: Device-to-radio link



Neighbor up and down signaling

Mobile Ad Hoc Networks (MANETs) are highly dynamic environments. Nodes might move into, or out of, radio range at a fast pace. Each time a node joins or leaves, the network topology must be logically reconstructed by the devices. Routing protocols normally use timer-driven hello messages or neighbor timeouts to track topology changes, but MANETs reliance on these mechanisms can result in unacceptably slow convergence.

The neighbor up/down signaling capability provides faster network convergence by using link-status signals generated by the radio. The radio notifies the device each time a link to another neighbor is established or terminated by the creation and termination of PPP over Ethernet (PPPoE) sessions. In the device, the routing protocols (Open Shortest Path First version 3 [OSPFv3] or Enhanced Interior Gateway Routing Protocol [EIGRP]) respond immediately to these signals by expediting formation of a new adjacency (for a new neighbor) or tearing down an existing adjacency (if a neighbor is lost). For example, if a vehicle drives behind a building and loses its connection, the device immediately senses the loss and establishes a new route to the vehicle through neighbors that are not blocked. This high-speed network convergence is essential for minimizing dropped voice calls and disruptions to video sessions.

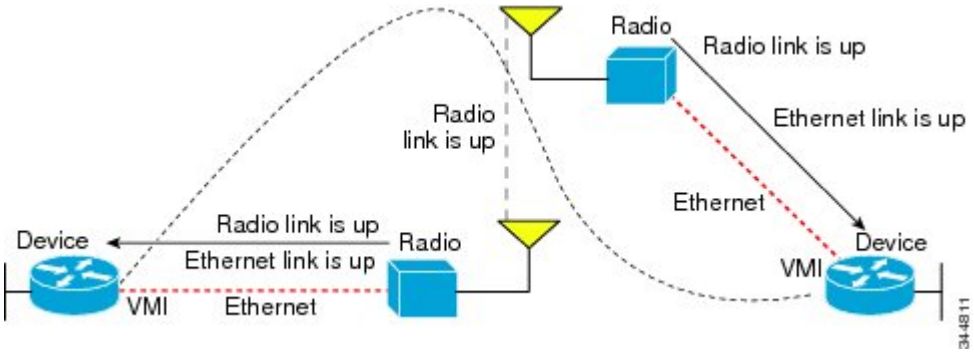
When virtual multipoint interfaces (VMIs) with PPPoE are used and a partner node has left or a new one has joined, the radio informs the device immediately of the topology change. Upon receiving the signal, the device immediately declares the change and updates the routing tables. The signaling capability provides these advantages:

- Reduces routing delays and prevents applications from timing out
- Enables network-based applications and information to be delivered reliably and quickly over directional radio links
- Provides faster convergence and optimal route selection so that delay-sensitive traffic such as voice and video are not disrupted
- Reduces impact on radio equipment by minimizing the need for internal queueing and buffering
- Provides consistent quality of service for networks with multiple radios

The messaging allows for flexible rerouting when necessary because of these factors:

- Fading of the radio links
- Congestion of the radio links
- Radio link power fade
- Utilization of the radio

Figure 9: Up and down signaling sequence



PPPoE credit-based and metric-based scaling and flow control

Each radio initiates a PPP over Ethernet (PPPoE) session with its local device as soon as the radio establishes a link to another radio. Once the PPPoE sessions are active for each node, a PPP session is then established end-to-end (device-to-device). This process is duplicated each time a radio establishes a new link.

The carrying capacity of each radio link might vary due to location changes or environmental conditions, and many radio transmission systems have limited buffering capabilities. To minimize the need for packet queueing in the radio, PPPoE protocol extensions enable the device to control traffic buffering in congestion situations. Implementing flow-control on these device-to-radio sessions allows use of quality of service (QoS) features such as fair queueing.

The flow-control solution utilizes a credit-granting mechanism documented in RFC 5578. When the PPPoE session is established, the radio can request a flow-controlled session. If the device acknowledges the request, all subsequent traffic must be flow controlled. If a flow-control session is requested and cannot be supported by the device, the session is terminated. Typically, both the radio and the device initially grant credits during session discovery. Once a device exhausts its credits, it must stop sending until additional credits are granted. Credits can be added incrementally over the course of a session.

Metrics scaling is used with high-performance radios that require high-speed links. The radio can express the maximum and current data rates with different scaler values. Credit scaling allows a radio to change the default credit grant (or scaling factor) of 64 bytes to its default value. You can display the maximum and current data rates and the scalar value set by the radio in the show VMI neighbor detail command output.

QoS provisioning on PPPoE extension session

This example describes QoS provisioning on PPPoE extension session.

```
policy-map rar_policer
 class class-default
  police 10000 2000 1000 conform-action transmit  exceed-action drop  violate-action drop
policy-map rar_shaper
 class class-default
  shape average percent 1

interface Virtual-Template2
 ip address 92.92.2.1 255.255.255.0
 no peer default ip address
 no keepalive
 service-policy input rar_policer
end
```

Configure the RAR feature in bypass mode

This task configures the RAR feature in bypass mode to provide radio-aware routing capabilities with Point-to-Point Protocol over Ethernet (PPPoE) services.

This configuration enables end-to-end RAR functionality in bypass mode. Before beginning the RAR configuration, you must first configure the **subscriber authorization enable** command to bring up the RAR session. Without enabling authorization, the Point-to-Point protocol does not recognize this as a RAR session and may not tag *manet_radio* in presentation of a PPPoE Active Discovery Initiate (PADI). By default, bypass mode does not appear in the configuration. It appears only if the mode is configured as bypass.

Procedure

Step 1 Configure a service for RAR.

Example:

```
policy-map type service rar-lab
 pppoe service manet_radio //note: Enter the pppoe service policy name as manet_radio
!
```

Step 2 Configure broadband settings.

Example:

```
bba-group pppoe VMI2
 virtual-template 2
service profile rar-lab
!
interface GigabitEthernet0/0/0
 description Connected to Client1
  negotiation auto
```

```
  pppoe enable group VMI2
!
```

Step 3 Configure the virtual template for bypass mode using one of the following options.

- **IP Address Configured under Virtual-Template Explicitly**

```
interface Virtual-Template2
ip address 192.168.90.3 255.255.255.0
no ip redirects
peer default ip address pool PPPoEpool2
ipv6 enable
ospfv3 1 network manet
ospfv3 1 ipv4 area 0
ospfv3 1 ipv6 area 0
no keepalive
service-policy input rar_policer Or/And
service-policy output rar_shaper
```

- **VMI Unnumbered Configured under Virtual Template**

```
interface Virtual-Template2
ip unnumbered vmi2
no ip redirects
peer default ip address pool PPPoEpool2
ipv6 enable
ospfv3 1 network manet
ospfv3 1 ipv4 area 0
ospfv3 1 ipv6 area 0
no keepalive
service-policy input rar_policer Or/And
service-policy output rar_shaper
```

Step 4 Configure the virtual multipoint interface in bypass mode.

Example:

```
interface vmi2 //configure the virtual multi interface
ip address 192.168.2.1 255.255.255.0
physical-interface GigabitEthernet0/0/0
mode bypass
```

Step 5 Configure OSPF routing.

Example:

```
router ospfv3 1
 router-id 1.1.1.1
!
 address-family ipv4 unicast
  redistribute connected metric 1 metric-type 1
  log-adjacency-changes
 exit-address-family
 !
 address-family ipv6 unicast
  redistribute connected metric-type 1
  log-adjacency-changes
 exit-address-family
```

```
!
ip local pool PPPoEpool2 192.168.2.3 192.168.2.254
```

The RAR feature is configured in bypass mode with PPPoE services, virtual template interfaces, and OSPF routing enabled for radio-aware routing functionality.

Verifying RAR Session Details

Use the provided show commands to monitor PPPoE session packets, VMI counters, neighbor details, and platform-specific hardware information for troubleshooting and verification.

```
Router#show pppoe session packets all
Total PPPoE sessions 2
session id: 9
local MAC address: 006b.f10e.a5e0, remote MAC address: 0050.56bc.424a
virtual access interface: Vi2.1, outgoing interface: Gi0/0/0
1646 packets sent, 2439363 received
176216 bytes sent, 117250290 received
PPPoE Flow Control Stats
Local Credits: 65535   Peer Credits: 65535   Local Scaling Value 64 bytes
Credit Grant Threshold: 28000    Max Credits per grant: 65535
Credit Starved Packets: 0
PADG xmit Seq Num: 32928     PADG Timer index: 0
PADG last rcvd Seq Num: 17313
PADG last nonzero Seq Num: 17306
PADG last nonzero rcvd amount: 2
PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
PADG xmit: 33308  rcvd: 17313
PADC xmit: 17313  rcvd: 19709
In-band credit pkt xmit: 7 rcvd: 2434422
Last credit packet snapshot
PADG xmit: seq_num = 32928, fcn = 0, bcn = 65535
PADC rcvd: seq_num = 32928, fcn = 65535, bcn = 65535
PADG rcvd: seq_num = 17313, fcn = 0, bcn = 65535
PADC xmit: seq_num = 17313, fcn = 65535, bcn = 65535
In-band credit pkt xmit: fcn = 61, bcn = 65533
In-band credit pkt rcvd: fcn = 0, bcn = 65534
==== PADQ Statistics ====
PADQ xmit: 0  rcvd: 0
session id: 10
local MAC address: 006b.f10e.a5e1, remote MAC address: 0050.56bc.7dcb
virtual access interface: Vi2.2, outgoing interface: Gi0/0/1
1389302 packets sent, 1852 received
77869522 bytes sent, 142156 received
PPPoE Flow Control Stats
Local Credits: 65535   Peer Credits: 65535   Local Scaling Value 64 bytes
Credit Grant Threshold: 28000    Max Credits per grant: 65535
Credit Starved Packets: 0
PADG xmit Seq Num: 18787     PADG Timer index: 0
PADG last rcvd Seq Num: 18784
PADG last nonzero Seq Num: 18768
PADG last nonzero rcvd amount: 2
PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
PADG xmit: 18787  rcvd: 18784
PADC xmit: 18784  rcvd: 18787
In-band credit pkt xmit: 1387764 rcvd: 956
Last credit packet snapshot
PADG xmit: seq_num = 18787, fcn = 0, bcn = 65535
PADC rcvd: seq_num = 18787, fcn = 65535, bcn = 65535
```

```
PADG rcvd: seq_num = 18784, fcn = 0, bcn = 65535
PADC xmit: seq_num = 18784, fcn = 65535, bcn = 65535
In-band credit pkt xmit: fcn = 0, bcn = 64222
In-band credit pkt rcvd: fcn = 0, bcn = 65534
==== PADQ Statistics ====
PADQ xmit: 0  rcvd: 1
```

```
Router#show pppoe session packets
Total PPPoE sessions 2
SID     Pkts-In         Pkts-Out        Bytes-In        Bytes-Out
9       2439391         1651            117252098       176714
10      1858            1389306         142580          77869914
```

```
Router#show vmi counters
Interface vmi2: - Last Clear Time =
Input Counts:
  Process Enqueue      =          0 (VMI)
  Fastswitch           =          0
  VMI Punt Drop:
       Queue Full      =          0
Output Counts:
  Transmit:
       VMI Process DQ  =       4280
       Fastswitch VA   =          0
       Fastswitch VMI  =          0
  Drops:
       Total           =          0
       QOS Error       =          0
       VMI State Error =          0
       Mcast NBR Error =          0
       Ucast NBR Error =          0
Interface vmi3: - Last Clear Time =
Input Counts:
  Process Enqueue      =          0 (VMI)
  Fastswitch           =          0
  VMI Punt Drop:
       Queue Full      =          0
Output Counts:
  Transmit:
       VMI Process DQ  =       2956
       Fastswitch VA   =          0
       Fastswitch VMI  =          0
  Drops:
       Total           =          0
       QOS Error       =          0
       VMI State Error =          0
       Mcast NBR Error =          0
       Ucast NBR Error =          0
Interface vmi4: - Last Clear Time =
Input Counts:
  Process Enqueue      =          0 (VMI)
  Fastswitch           =          0
  VMI Punt Drop:
       Queue Full      =          0
Output Counts:
  Transmit:
       VMI Process DQ  =          0
       Fastswitch VA   =          0
       Fastswitch VMI  =          0
  Drops:
       Total           =          0
       QOS Error       =          0
       VMI State Error =          0
       Mcast NBR Error =          0
```

```
       Ucast NBR Error =          0
Router#

Router#show vmi neighbor details
1 vmi2 Neighbors
1 vmi3 Neighbors
0 vmi4 Neighbors
2 Total Neighbors
vmi2   IPV6 Address=FE80::21E:E6FF:FE43:F500
       IPV6 Global Addr=::
       IPV4 Address=92.92.2.2, Uptime=05:15:01
       Output pkts=89, Input pkts=0
       No Session Metrics have been received for this neighbor.
       Transport PPPoE, Session ID=9
       INTERFACE STATS:
          VMI Interface=vmi2,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          V-Access intf=Virtual-Access2.1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          Physical intf=GigabitEthernet0/0/0,
             Input qcount=0, drops=0, Output qcount=0, drops=0
PPPoE Flow Control Stats
 Local Credits: 65535   Peer Credits: 65535   Local Scaling Value 64 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 33038     PADG Timer index: 0
 PADG last rcvd Seq Num: 17423
 PADG last nonzero Seq Num: 17420
 PADG last nonzero rcvd amount: 2
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 33418  rcvd: 17423
 PADC xmit: 17423  rcvd: 19819
 In-band credit pkt xmit: 7 rcvd: 2434446
 Last credit packet snapshot
  PADG xmit: seq_num = 33038, fcn = 0, bcn = 65535
  PADC rcvd: seq_num = 33038, fcn = 65535, bcn = 65535
  PADG rcvd: seq_num = 17423, fcn = 0, bcn = 65535
  PADC xmit: seq_num = 17423, fcn = 65535, bcn = 65535
  In-band credit pkt xmit: fcn = 61, bcn = 65533
  In-band credit pkt rcvd: fcn = 0, bcn = 65534
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 0
vmi3   IPV6 Address=FE80::21E:7AFF:FE68:6100
       IPV6 Global Addr=::
       IPV4 Address=91.91.91.4, Uptime=05:14:55
       Output pkts=6, Input pkts=0
       METRIC DATA: Total rcvd=1, Avg arrival rate (ms)=0
          CURRENT: MDR=128000 bps, CDR=128000 bps
                   Lat=0 ms, Res=100, RLQ=100, load=0
          MDR      Max=128000 bps, Min=128000 bps, Avg=128000 bps
          CDR      Max=128000 bps, Min=128000 bps, Avg=128000 bps
          Latency  Max=0, Min=0, Avg=0 (ms)
          Resource Max=100%, Min=100%, Avg=100%
          RLQ      Max=100, Min=100, Avg=100
          Load     Max=0%, Min=0%, Avg=0%
       Transport PPPoE, Session ID=10
       INTERFACE STATS:
          VMI Interface=vmi3,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          V-Access intf=Virtual-Access2.2,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          Physical intf=GigabitEthernet0/0/1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
PPPoE Flow Control Stats
 Local Credits: 65535   Peer Credits: 65535   Local Scaling Value 64 bytes
```

```
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 18896     PADG Timer index: 0
 PADG last rcvd Seq Num: 18894
 PADG last nonzero Seq Num: 18884
 PADG last nonzero rcvd amount: 2
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 18896  rcvd: 18894
 PADC xmit: 18894  rcvd: 18896
 In-band credit pkt xmit: 1387764 rcvd: 961
 Last credit packet snapshot
  PADG xmit: seq_num = 18896, fcn = 0, bcn = 65535
  PADC rcvd: seq_num = 18896, fcn = 65535, bcn = 65535
  PADG rcvd: seq_num = 18894, fcn = 0, bcn = 65535
  PADC xmit: seq_num = 18894, fcn = 65535, bcn = 65535
  In-band credit pkt xmit: fcn = 0, bcn = 64222
  In-band credit pkt rcvd: fcn = 0, bcn = 65534
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 1

Router#show vmi neighbor details vmi 2
              1 vmi2 Neighbors
vmi2   IPV6 Address=FE80::21E:E6FF:FE43:F500
       IPV6 Global Addr=::
       IPV4 Address=92.92.2.2, Uptime=05:16:03
       Output pkts=89, Input pkts=0
       No Session Metrics have been received for this neighbor.
       Transport PPPoE, Session ID=9
       INTERFACE STATS:
          VMI Interface=vmi2,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          V-Access intf=Virtual-Access2.1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          Physical intf=GigabitEthernet0/0/0,
             Input qcount=0, drops=0, Output qcount=0, drops=0
PPPoE Flow Control Stats
 Local Credits: 65535   Peer Credits: 65535   Local Scaling Value 64 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 33100     PADG Timer index: 0
 PADG last rcvd Seq Num: 17485
 PADG last nonzero Seq Num: 17449
 PADG last nonzero rcvd amount: 2
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 33480  rcvd: 17485
 PADC xmit: 17485  rcvd: 19881
 In-band credit pkt xmit: 7 rcvd: 2434460
 Last credit packet snapshot
  PADG xmit: seq_num = 33100, fcn = 0, bcn = 65535
  PADC rcvd: seq_num = 33100, fcn = 65535, bcn = 65535
  PADG rcvd: seq_num = 17485, fcn = 0, bcn = 65535
  PADC xmit: seq_num = 17485, fcn = 65535, bcn = 65535
  In-band credit pkt xmit: fcn = 61, bcn = 65533
  In-band credit pkt rcvd: fcn = 0, bcn = 65534
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 0

Router#show platform hardware qfp active feature ess session
Current number sessions: 2
Current number TC flow: 0
Feature Type: A=Accounting D=Policing(DRL) F=FFR M=DSCP Marking L=L4redirect P=Portbundle
T=TC
   Session    Type      Segment1      SegType1      Segment2      SegType2 Feature Other
-------------------------------------------------------------------------------------
```

```
        21     PPP 0x0000001500001022    PPPOE 0x0000001500002023    LTERM -------
        24     PPP 0x0000001800003026    PPPOE 0x0000001800004027    LTERM -------

Router#show platform software subscriber pppoe_fctl evsi 21
PPPoE Flow Control Stats
 Local Credits: 65535   Peer Credits: 65535   Local Scaling Value 64 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 33215     PADG Timer index: 0
 PADG last rcvd Seq Num: 17600
 PADG last nonzero Seq Num: 17554
 PADG last nonzero rcvd amount: 2
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 33595  rcvd: 17600
 PADC xmit: 17600  rcvd: 19996
 In-band credit pkt xmit: 7 rcvd: 2434485
 Last credit packet snapshot
  PADG xmit: seq_num = 33215, fcn = 0, bcn = 65535
  PADC rcvd: seq_num = 33215, fcn = 65535, bcn = 65535
  PADG rcvd: seq_num = 17600, fcn = 0, bcn = 65535
  PADC xmit: seq_num = 17600, fcn = 65535, bcn = 65535
  In-band credit pkt xmit: fcn = 61, bcn = 65533
  In-band credit pkt rcvd: fcn = 0, bcn = 65534
BQS buffer statistics
 Current packets in BQS buffer: 0
 Total en-queue packets: 0 de-queue packets: 0
 Total dropped packets: 0
Internal flags: 0x0

Router#show platform hardware qfp active feature ess session id 21
Session ID: 21
  EVSI type: PPP
  SIP Segment ID: 0x1500001022
  SIP Segment type: PPPOE
  FSP Segment ID: 0x1500002023
  FSP Segment type: LTERM
  QFP if handle: 16
  QFP interface name: EVSI21
  SIP TX Seq num: 0
  SIP RX Seq num: 0
  FSP TX Seq num: 0
  FSP RX Seq num: 0
  Condition Debug: 0x00000000
    session

Router#show ospfv3 neighbor
          OSPFv3 1 address-family ipv4 (router-id 3.3.3.3)
Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
1.1.1.1           0   FULL/  -        00:01:32    19              Virtual-Access2.1
          OSPFv3 1 address-family ipv6 (router-id 3.3.3.3)
Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
1.1.1.1           0   FULL/  -        00:01:52    19              Virtual-Access2.1
Router#

Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
a - application route
+ - replicated route, % - next hop override, p - overrides from PfR
Gateway of last resort is not set
      90.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
```

```
C        90.90.90.0/24 is directly connected, Virtual-Access2.1
O        90.90.90.4/32 [110/1] via 90.90.90.4, 00:00:03, Virtual-Access2.1
L        90.90.90.5/32 is directly connected, Virtual-Access2.1
      92.0.0.0/32 is subnetted, 1 subnets
C        92.92.2.21 is directly connected, Virtual-Access2.1
```

DLEP

Cisco Dynamic Link Exchange Protocol (DLEP) is a protocol in the Radio Aware Routing (RAR) family that addresses the challenges faced when merging IP routing and radio frequency (RF) communications.

DLEP capabilities

Cisco provides capabilities that enable:

- Optimal route selection based on feedback from radios
- Faster convergence when nodes join and leave the network
- Efficient integration of point-to-point, point-to-multipoint and broadcast multi-access radio topologies with multi-hop routing
- Flow-controlled communications between the radio and its partner router using rate-based Quality of Service (QoS) policies
- Dynamic shaping of fluctuating RF bandwidth in near real time to provide optimized use of actual RF bandwidth

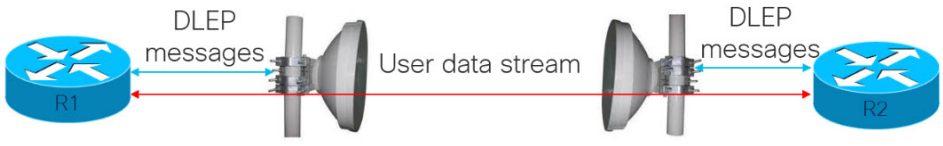
DLEP topology and packet flow

DLEP topology and packet flow is a communication mechanism that

- enables radios to communicate link quality information to routers in real-time
- allows routing metrics to be adjusted dynamically to favor the best path, and
- supports multiple packet flow scenarios through virtual interfaces in IOS-XE platforms.

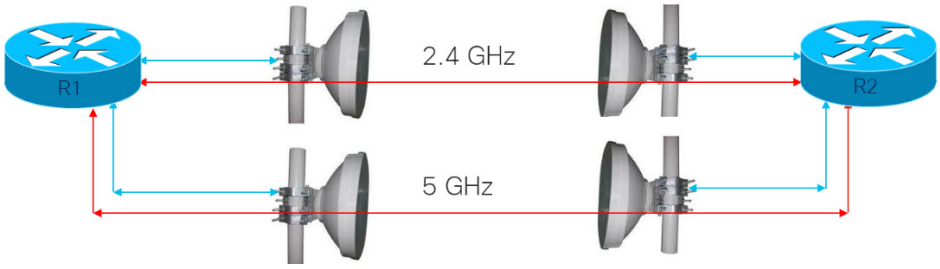
DLEP topology characteristics

The DLEP exchange between the router and radio allows the radio to tell the router about the link quality. This is somewhat analogous to the way the bar icon on your cell phone tells you about your Wi-Fi or LTE signal quality.



With DLEP, we can make use of routing distances with equal cost, where metrics are updated based in real time based on best path.

Without DLEP, there are two equal cost paths to any unadjusted routing protocol. With DLEP, routing metrics can be adjusted in real-time to favor the best path.



Note Atmospheric conditions and interference will ultimately favor one band versus the other.

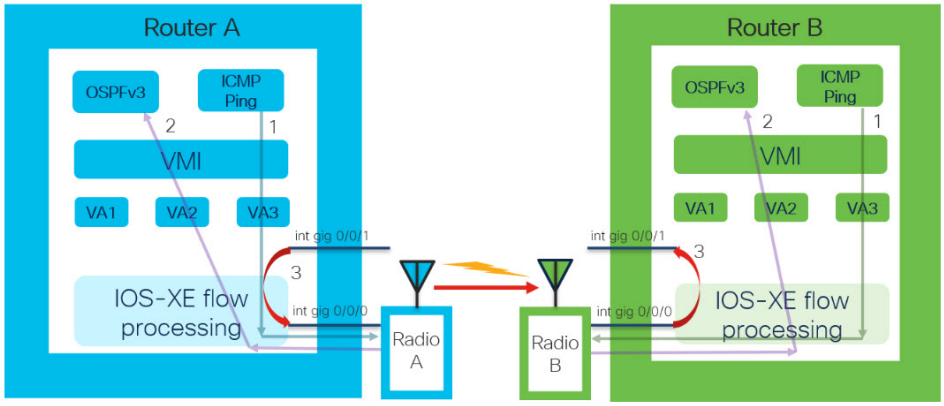
DLEP packet flow includes three different scenarios:

- Packet sent from IOS-XE to the DLEP neighbor (radio)
- Packet sent from DLEP neighbor (radio) to IOS-XE
- Non-DLEP traffic passing between the ESR-6300 Layer 3 (WAN) interfaces, for example, user traffic

IOS-XE platform interfaces include:

- Virtual Multipoint Interface (VMI): The VMI interface acts as an umbrella interface for all virtual access interfaces, which is used for routing protocols such as OSPFv3 and EIGRP. Routing protocols, for example OSPFv3, see a single VMI interface instead of all VA interfaces. This helps reduce routing table size without impacting the integrity of network.
- Virtual Template (VT): Virtual Template serves as the template for every Virtual Access interface.
- Virtual Access (VA) interface: One VA is created for each DLEP Neighbor (Radio) discovered.
- Underlaying physical Layer 3 WAN interface (Gi 0/0/0 and Gi 0/0/1), or even sub-interface (Gi 0/0/0.2 or Gi0/0/1.2

This diagram describes the packet flow:



Item	Description
1	Packet sent from IOS-XE to DLEP neighbor (to Radio): packet with DST MAC of neighbor MAC, neighbor IP sent out from DLEP Physical interface.
2	DLEP Packet (from radio) received thru DLEP Physical interface to IOS-XE: packet needs to be delivered to IOS routing protocol marked as from VMI interface.
3	End to End user data.

DLEP communication analogy

The DLEP exchange between the router and radio is somewhat analogous to the way the bar icon on your cell phone tells you about your Wi-Fi or LTE signal quality.

License requirements

The Cisco Dynamic Link Exchange Protocol (DLEP) feature requires the Network Advantage license.

Configuring DLEP with OSPFv3

Configuring DLEP with OSPFv3 is a network configuration process that

- initiates, verifies, and manages all aspects of Dynamic Link Exchange Protocol (DLEP) on an interface
- requires configuration of physical interfaces, VMI interfaces, and Virtual Templates, and
- must include VMI and Virtual-Template configuration before attaching a Virtual-Template to a physical interface.

DLEP interface configuration requirements

DLEP uses these interfaces that need to be configured:

- Physical interface
- VMI interface
- Virtual Templates



Important Before attaching a Virtual-Template to a physical interface, you need to configure the VMI and Virtual-Template. See this example of the error you receive:

```
Router(config)#int gi0/0/0
Router(config-if)#ip dlep vtemplate 1
DLEP: ERROR. A valid VMI and Virtual Template MUST be present before configurating ip dlep
DLEP: ERROR. An Interface IP address MUST be specified for DLEP.
```

Configure the virtual multipoint interface

Use this task to configure virtual multipoint interfaces (VMIs) to aggregate the virtual access interfaces created by DLEP sessions and to enable routing protocol support.

By default, VMIs operate in aggregate mode, which means that all the virtual access interfaces created by DLEP sessions are aggregated logically under the configured VMI. Applications above Layer 2, such as Enhanced Interior Gateway Routing Protocol (EIGRP) and Open Shortest Path First version 3 (OSPFv3), must be defined only on the VMI. Packets sent to the VMI are forwarded to the correct virtual access interface.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
                        Enter configuration commands, one per line.  End with CNTL/Z.
                        Router(config)#
```

Step 2 Use the **interface vmi** *number* command to create a VMI and enter interface configuration mode.

This example creates vmi1.

Example:

```
Router(config)# interface vmi 1
                        Router(config-if)#
```

Step 3 Use the **ip unnumbered** *interface* command to configure the VMI to use the IP address of the physical interface.

Example:

```
Router(config-if)# ip unnumbered gigabitEthernet 0/0/0
```

Step 4 Use the **physical-interface** *interface* command to bind the physical interface to the VMI for packet flow.

Example:

```
Router(config-if)# physical-interface gigabitEthernet 0/0/0
```

Step 5 Use the **ipv6 enable** command to enable IPv6 support on the VMI.

OSPFv3 IPv4 requires IPv6 support to be enabled at the interface level.

Example:

```
Router(config-if)# ipv6 enable
```

Step 6 Use the **ospfv3** *process-id* **ipv4 area** *area-id* command to enable the VMI to participate in OSPFv3 or EIGRP routing.

Example:

```
Router(config-if)# ospfv3 1 ipv4 area 0
```

Step 7 Use the **exit** command to exit interface configuration mode and return to global configuration mode.

Example:

```
Router(config-if)# exit
Router(config)#
```

Step 8 Use the **router ospfv3** *process-id* command to enter the global configuration for OSPFv3.

Example:

```
Router(config)# router ospfv3 1
```

Step 9 Use the **address-family ipv4 unicast** command to add the address family for IPv4 unicast routing under the global OSPFv3 configuration.

Example:

```
Router(config-router)# address-family ipv4 unicast
Router(config-router-af)#
```

The virtual multipoint interface is configured and enabled for routing protocols, allowing DLEP sessions to create virtual access interfaces that are aggregated under the VMI.

Configure the virtual template

Use this task to create the DLEP virtual template that enables Dynamic Link Exchange Protocol (DLEP) functionality on the router interface. DLEP configuration requires a virtual template to be defined for proper operation.

Procedure

Step 1 Use the **enable** command to enter the privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
Router#
```

Step 2 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

Step 3 Use the **interface Virtual-Template** *number* command to create the virtual template interface and enter interface configuration mode.

You must use the same virtual template interface when you configure DLEP on the physical interface.

Example:

```
Router(config)# interface Virtual-Template 1
Router(config-if)#
```

Step 4 Use the **ip unnumbered** *interface* command to configure the virtual template interface to use the IP address of the physical interface.

Example:

```
Router(config-if)# ip unnumbered gigabitEthernet 0/0/0
```

Step 5 Use the **ipv6 enable** command to enable IPv6 support on the virtual template interface.

OSPFv3 IPv4 requires IPv6 support to be enabled at the interface level.

Example:

```
Router(config-if)# ipv6 enable
```

Step 6 Use the **exit** command to exit interface configuration mode and return to global configuration mode.

Example:

```
Router(config-if)# exit
Router(config)#
```

The virtual template interface is configured and ready for DLEP configuration on the physical interface.

Configure the physical interface

Before you begin

Before you attach a DLEP template to a WAN interface, configure a physical IP address and create the VMI and virtual template interfaces.

Use this task to enable Dynamic Link Exchange Protocol (DLEP) functionality on a physical WAN interface by applying the necessary configuration commands and attaching a DLEP template.

DLEP configuration is currently supported on the WAN interface of IOS XE platforms. You must configure both the VMI and the virtual template interface before you configure the physical interface. You can attach the DLEP configuration to the WAN interface in several ways. The following are some examples:

- DLEP template with a well-known IP address (recommended)
- DLEP template with TCP or UDP ports configured between the server (router) and the client (radio)
- DLEP template with a dynamic port on the server (router)
- DLEP template attached in discovery mode

For each of these four modes, you also have the option to enable Generalized TTL Security Mechanism (GTSM).

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

Step 2 Use the **interface** *type number* command to enter interface configuration mode for the WAN interface.

Example:

```
Router(config)# interface gigabitEthernet 0/0/0
Router(config-if)#
```

Step 3 Use the **ipv6 enable** command to enable IPv6 support at the interface level.

Example:

```
Router(config-if)# ipv6 enable
```

Step 4 Use the **ip address** *ip-address subnet-mask* command to assign a physical IP address to the WAN interface.

Example:

```
Router(config-if)# ip address 10.0.0.1 255.255.255.0
```

Step 5 Use the **ip dlep vtemplate** *number* command to attach the DLEP template to the WAN interface in discovery mode.

The other configuration examples are described separately.

Example:

```
Router(config-if)# ip dlep vtemplate 1
```

Step 6 Use the **no shutdown** command to bring up the interface.

Example:

```
Router(config-if)# no shutdown
```

Step 7 Use the **exit** command to exit interface configuration mode and return to global configuration mode.

Example:

```
Router(config-if)# exit
```

The physical interface is configured with DLEP support, and the DLEP template is attached to the WAN interface. The interface is enabled and ready for DLEP communication.

DLEP configuration modes

DLEP configuration modes are attachment methods that enable DLEP virtual templates to be connected to different router interface types in various operational configurations.

Configuration requirements

DLEP virtual templates can be attached in different modes to the WAN or sub-interface of the router.

Note You must have VMI and Virtual-Templates already configured prior to continuing with any of the configuration modes in this section.

Configure the DLEP client and server based on port number

Use this task to configure the DLEP server and client UDP and TCP ports to enable proper communication between the router and radio devices with custom port assignments.

DLEP (Dynamic Link Exchange Protocol) is used for dynamic link management between routers and radio devices. By default, DLEP uses standard ports, but you may need to configure custom ports for specific network requirements or to avoid port conflicts.

In this example, you configure the DLEP server, and the client UDP and TCP ports.

Procedure

Step 1 Use these steps to configure the interface with IP addressing and DLEP parameters:

a) Use the **interface** *type number* command to enter interface configuration mode for the specified interface.

Example:

```
Router(config)# interface gi0/0/0
```

b) Use the **ip address** *ip-address subnet-mask* command to assign an IP address and subnet mask to the interface.

Example:

```
Router(config-if)# ip address 10.0.0.1 255.255.255.0
```

c) Use the **ipv6 enable** command to enable IPv6 processing on the interface.

Example:

```
Router(config-if)# ipv6 enable
```

d) Use the **ip dlep vtemplate** *number* **port** *udp-port* **tcp port** *tcp-port* **client ip** *ip-address* **port** *client-port* command to attach the DLEP virtual template to the interface and specify the server and client port numbers.

Example:

```
Router(config-if)# ip dlep vtemplate 1 port 11113 tcp port 11114 client ip 10.0.0.3 port
 11115
```

e) Use the **no shutdown** command to enable the interface.

Example:

```
Router(config-if)# no shutdown
```

In this DLEP configuration, the server (router) listens on UDP port 11113 and TCP port 11114, and the client (radio) listens on TCP port 11115. The UDP port of the client is 854 by default.

Step 2 Use the **show dlep clients** command to validate the configuration.

Example:

```
Router# show dlep clients
DLEP Clients for all interfaces:

DLEP Clients for Interface GigabitEthernet0/0/0
DLEP Server IP=10.0.0.1:11113 Sock=0

DLEP Client IP=10.0.0.3:11115 TCP Socket fd=1
Peer ID=1, Virtual template=1
Description: DLEP-Radio1-Path-1
Peer Timers (all values in milliseconds):
Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
Neighbor Timers (all values in seconds):
Activity timeout=0, Neighbor Down ACK=10

Supported Metrics:
Link RLQ RX Metric : 100
Link RLQ TX Metric : 100
```

```
Link Resources Metric : 100
Link MTU Metric : 100
Link Latency Metric : 250 microseconds
Link CDR RX Metric : 100000000 bps
Link CDR TX Metric : 100000000 bps
Link MDR RX Metric : 100000000 bps
Link MDR TX Metric : 100000000 bps
```

The DLEP server and client are configured with custom port numbers, and the configuration is validated, displaying the established connection and supported metrics between the router and radio device.

Configure DLEP with dynamic port on server (router)

Configure DLEP server and client communication settings to establish proper connectivity between the router (server) and radio (client) with specific port configurations.

In this configuration, you are setting up the DLEP server where the server (router) listens on default UDP/TCP ports while the client (radio) listens on a specific TCP port.

Procedure

Step 1 Configure the interface with IP address and DLEP settings.

Example:

```
Router(config)#interface gi0/0/0
Router(config-if)# ip address 10.0.0.1 255.255.255.0
Router(config-if)# ipv6 enable
Router(config-if)# ip dlep vtemplate 1 client ip 10.0.0.3 port 11115
Router(config-if)# no shutdown
```

DLEP configuration where server (router) is listening to default UDP/TCP ports. Whereas client (radio) is listening to TCP port 11115. UDP port of client is by default on 854.

Step 2 Validate the configuration using the **show DLEP clients** command.

Example:

```
Router# show dlep clients
DLEP Clients for all interfaces:

DLEP Clients for Interface GigabitEthernet0/0/0
DLEP Server IP=10.0.0.1:55555 Sock=0

DLEP Client IP=10.0.0.3:11115 TCP Socket fd=1
Peer ID=1, Virtual template=1
 Description: DLEP-Radio1-Path-1
Peer Timers (all values in milliseconds):
Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
Neighbor Timers (all values in seconds):
Activity timeout=0, Neighbor Down ACK=10

Supported Metrics:
Link RLQ RX Metric : 100
Link RLQ TX Metric : 100
Link Resources Metric : 100
Link MTU Metric : 100
```

```
Link Latency Metric : 250 microseconds
Link CDR RX Metric : 100000000 bps
Link CDR TX Metric : 100000000 bps
Link MDR RX Metric : 100000000 bps
Link MDR TX Metric : 100000000 bps
```

Step 3 Validate the configuration using the **show DLEP counters** command.

Example:

```
Router# show dlep counters
DLEP Counters for GigabitEthernet0/0/0

Last Clear Time =

DLEP Version = RFC 8175
DLEP Server IP=10.0.0.1:55555
DLEPv5 TCP Port = 55556

Peer Counters:
RX Peer Discovery     0      TX Peer Offer             0
RX Peer Offer         0      TX Peer Discovery         0
RX Peer Init          0      TX Peer Init Ack          0
RX Peer Init Ack      1      TX Peer Init              1
RX Heartbeat          58     TX Heartbeat              58
RX Peer Terminate     0      TX Peer Terminate Ack     0
RX Peer Terminate Ack 0      TX Peer Terminate         0

Neighbor Counters:
RX Neighbor Up        0      TX Neighbor Up Ack        0
RX Metric             0
RX Neighbor Down      0      TX Neighbor Down Ack      0
RX Neighbor Down Ack  0      TX Neighbor Down          0

Exception Counters:
RX Invalid Message    0      RX Unknown Message        0
Pre-Existing Neighbor 0      Neighbor Resource Error   0
Neighbor Not Found    0      Neighbor Msg Peer Not Up  0

Timer Counters:
Peer Heartbeat Timer         58
Peer Terminate Ack Timer     0
Neighbor Terminate Ack Timer 0
Neighbor Activity Timer      0
Radio Connect Timer          5

Single Timer Wheel "Manet Infra Wheel"
Granularity      = 250 msec
Wheel size       = 4096
Spoke index      = 3592
Tick count       = 3592
Flags            = 0x00
Active timers    = 1
High water mark  = 2
Started timers   = 164
Restarted timers = 2
Cancelled timers = 3
Expired timers   = 158
Long timers      = 0
Long timer revs  = 0
Timer suspends   = 0
```

The DLEP configuration is complete and validated. The server (router) is listening on default UDP/TCP ports while communicating with the client (radio) on the specified TCP port 11115. The show commands confirm proper connectivity and display DLEP metrics and counters.

Attach the DLEP template in discovery mode

Use this task to configure DLEP server operation in discovery mode, enabling it to send Peer Discovery signals and establish communication with DLEP clients.

When in discovery mode, the DLEP server sends out Peer Discovery signals and waits for a Peer Offer signal from the radio.

Procedure

Step 1 Use these steps to configure the interface with the DLEP template and enable discovery mode:

a) Use the **interface** *type number* command to enter interface configuration mode for the specified interface.

Example:

```
Router(config)# interface gi0/0/0
```

b) Use the **ip address** *ip-address subnet-mask* command to assign an IP address and subnet mask to the interface.

Example:

```
Router(config-if)# ip address 10.0.0.1 255.255.255.0
```

c) Use the **ipv6 enable** command to enable IPv6 processing on the interface.

Example:

```
Router(config-if)# ipv6 enable
```

d) Use the **ip dlep vtemplate** *number* command to attach the DLEP virtual template to the interface.

Example:

```
Router(config-if)# ip dlep vtemplate 1
```

e) Use the **no shutdown** command to enable the interface.

Example:

```
Router(config-if)# no shutdown
```

In this DLEP configuration, the server (router) listens on the default UDP and TCP ports.

Note
To receive the peer discovery messages, you must explicitly configure the UDP port on the client to 55555.

Step 2 Validate the configuration using the **show dlep clients** and **show dlep counters** commands.

a) Use the **show dlep clients** command to display the DLEP client connections.

Example:

```
Router# show dlep clients
DLEP Clients for all interfaces:

DLEP Clients for Interface GigabitEthernet0/0/0
```

```
DLEP Server IP=10.0.0.1:55555 Sock=0

DLEP Client IP=10.0.0.3:11115 TCP Socket fd=1
 Peer ID=1, Virtual template=1
 Description: DLEP-Radio1-Path-1
 Peer Timers (all values in milliseconds):
  Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
 Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

 Supported Metrics:
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
  Link MTU Metric : 100
  Link Latency Metric : 250 microseconds
  Link CDR RX Metric : 100000000 bps
  Link CDR TX Metric : 100000000 bps
  Link MDR RX Metric : 100000000 bps
  Link MDR TX Metric : 100000000 bps
```

b) Use the **show dlep counters** command to verify peer communication.

Example:

```
Router# show dlep counters
DLEP Counters for GigabitEthernet0/0/0

Last Clear Time =

DLEP Version = RFC 8175
DLEP Server IP=10.0.0.1:55555
DLEPv5 TCP Port = 55556

Peer Counters:
RX Peer Discovery     0      TX Peer Offer             0
RX Peer Offer         3      TX Peer Discovery         194
RX Peer Init          0      TX Peer Init Ack          0
RX Peer Init Ack      3      TX Peer Init              3
RX Heartbeat          710    TX Heartbeat              707
RX Peer Terminate     0      TX Peer Terminate Ack     0
RX Peer Terminate Ack 0      TX Peer Terminate         2

Neighbor Counters:
RX Neighbor Up        0      TX Neighbor Up Ack        0
RX Metric             0
RX Neighbor Down      0      TX Neighbor Down Ack      0
RX Neighbor Down Ack  0      TX Neighbor Down          0

Exception Counters:
RX Invalid Message    0      RX Unknown Message        0
Pre-Existing Neighbor 0      Neighbor Resource Error   0
Neighbor Not Found    0      Neighbor Msg Peer Not Up  0

Timer Counters:
Peer Heartbeat Timer         709
Peer Terminate Ack Timer     2
Neighbor Terminate Ack Timer 0
Neighbor Activity Timer      0
Radio Connect Timer          3

Single Timer Wheel "Manet Infra Wheel"
Granularity      = 250 msec
Wheel size       = 4096
Spoke index      = 8
```

```
Tick count       = 24584
Flags            = 0x00
Active timers    = 1
High water mark  = 2
Started timers   = 1209
Restarted timers = 4
Cancelled timers = 14
Expired timers   = 1190
Long timers      = 0
Long timer revs  = 0
Timer suspends   = 0
```

The DLEP server is successfully configured in discovery mode and can communicate with DLEP clients. The verification commands display active client connections and communication statistics.

Use a DLEP template with a well-known IP address

Use this task to configure DLEP with a well-known IP address to enable standardized multicast communication between routers and radios using the RFC 8175 specification.

DLEP is based on RFC 8175 and uses the well-known IP address 224.0.0.117 on the server to communicate with radios. Additional prerequisites must be enabled for this configuration to work.

Procedure

Step 1 Use the **ip multicast-routing distributed** command to enable multicast routing on the router.

Example:

```
Router(config)# ip multicast-routing distributed
```

Step 2 Use these steps to configure the IPv4 address of the server:

a) Use the **interface** *type number* command to enter interface configuration mode.

Example:

```
Router(config)# interface gi0/0/0
```

b) Use the **ip address** *ip-address subnet-mask* command to assign an IP address and subnet mask to the interface.

Example:

```
Router(config-if)# ip address 10.1.2.3 255.255.255.0
```

c) Use the **ipv6 enable** command to enable IPv6 support on the interface.

Example:

```
Router(config-if)# ipv6 enable
```

d) Use the **no shutdown** command to bring up the interface.

Example:

```
Router(config-if)# no shutdown
```

Step 3 Use the **ip pim sparse-dense-mode** command to enable sparse-dense mode.

Example:

```
Router(config-if)# ip pim sparse-dense-mode
```

Step 4 Use these commands to enable the router to forward multicast traffic:

Example:

```
Router(config-if)# ip mfib cef in
                      Router(config-if)# ip mfib cef out
                      Router(config-if)# ip mfib forwarding in
                      Router(config-if)# ip mfib forwarding out
```

Step 5 Use the **ip dlep vtemplate** *number* **well-known ip** *ip-address* command to enable the DLEP virtual template to listen to multicast traffic.

Note
In the output of the **show running-config** command, this configuration is displayed as `ip dlep vtemplate 1`, which means that `well-known ip 224.0.0.117` is hidden.

Example:

```
Router(config-if)# ip dlep vtemplate 1 well-known ip 224.0.0.117
```

Step 6 Use the **show dlep clients** command to validate the configuration.

Example:

```
Router# show dlep clients

                      DLEP Clients for all interfaces:
                      DLEP Clients for Interface GigabitEthernet0/0/0
                      DLEP Server IP=10.1.2.3:55555 Sock=2

                      DLEP Client IP=10.1.2.4:854 TCP Socket fd=3
                      Peer ID=1, Virtual template=1
                      Description: OONF DLEP Radio
                      Peer Timers (all values in milliseconds):
                      Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
                      Neighbor Timers (all values in seconds):
                      Activity timeout=0, Neighbor Down ACK=10

                      Supported Metrics:
                      Link RLQ RX Metric : 100
                      Link RLQ TX Metric : 100
                      Link Resources Metric : 100
                      Link Latency Metric : 1000 microseconds
                      Link CDR RX Metric : 104857600 bps
                      Link CDR TX Metric : 104857600 bps
                      Link MDR RX Metric : 104857600 bps
                      Link MDR TX Metric : 104857600 bps
```

The DLEP template is configured with the well-known IP address, and multicast routing is enabled for communication with radios.

DLEP configuration with GTSM

Generalized TTL Security Mechanism (GTSM) can be used to provide an additional layer of security for the DLEP session.

Table 21: DLEP configuration modes

DLEP Configuration Mode	Command to attach DLEP template under WAN or sub-interface
DLEP template with TCP/UDP port based between server and client.	`Router(config)# interface gi0/0/0` `Router(config-if)# ip dlep vtemplate 1 port 11113 tcp port 11114 gtsm client ip 10.0.0.3 port 11115`
DLEP configuration with dynamic port on server.	`Router(config)# interface gi0/0/0` `Router(config-if)# ip dlep vtemplate 1 gtsm client ip 10.0.0.3 port 11115`
DLEP template attach in discovery mode.	`Router(config)# interface gi0/0/0` `Router(config-if)# ip dlep vtemplate 1 gtsm`
DLEP template with well-known ip address.	`Router(config)# interface gi0/0/0` `Router(config-if)# ip dlep vtemplate 1 gtsm well-known ip 224.0.0.117`

Configure DLEP with OSPFv3

Use this task to configure Dynamic Link Exchange Protocol (DLEP) with OSPFv3 to establish DLEP communication between routers and radio devices, enabling adaptive routing based on link characteristics.

Note DLEP works only with the OSPFv3 or EIGRP routing protocols. This task uses OSPFv3.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ipv6 unicast-routing** command to enable IPv6 unicast routing.

OSPFv3 IPv4 requires IPv6 support to be enabled at the interface level.

Example:

```
Router(config)# ipv6 unicast-routing
```

Step 3 Use these steps to configure the virtual multipoint interface (VMI):

a) Use the **interface vmi** *number* command to create the VMI and enter interface configuration mode.

Example:

```
Router(config)# interface vmi 1
Router(config-if)#
```

b) Use the **ip unnumbered** *interface* command to configure the VMI to use the IP address of the physical interface.

Example:

```
Router(config-if)# ip unnumbered GigabitEthernet 0/0/0
```

c) Use the **physical-interface** *interface* command to bind the physical interface to the VMI for packet flow.

Example:

```
Router(config-if)# physical-interface GigabitEthernet 0/0/0
```

d) Use the **ipv6 enable** command to enable IPv6 support on the VMI.

Example:

```
Router(config-if)# ipv6 enable
```

e) Use the **ospfv3** *process-id* **ipv4 area** *area-id* command to enable the VMI to participate in OSPFv3 routing.

Example:

```
Router(config-if)# ospfv3 1 ipv4 area 0
```

f) Use the **exit** command to return to global configuration mode.

Example:

```
Router(config-if)# exit
```

Step 4 Use these steps to configure the virtual template:

a) Use the **interface Virtual-Template** *number* command to create the virtual template interface and enter interface configuration mode.

You must use the same virtual template interface when you attach it to the physical interface.

Example:

```
Router(config)# interface Virtual-Template 1
Router(config-if)#
```

b) Use the **ip unnumbered** *interface* command to configure the virtual template interface to use the IP address of the physical interface.

Example:

```
Router(config-if)# ip unnumbered GigabitEthernet 0/0/0
```

c) Use the **ipv6 enable** command to enable IPv6 support on the virtual template interface.

Example:

```
Router(config-if)# ipv6 enable
```

d) Use the **exit** command to return to global configuration mode.

Example:

```
Router(config-if)# exit
Router(config)#
```

Step 5 Use these steps to configure the WAN interface and attach the DLEP template:

a) Use the **interface** *type number* command to enter interface configuration mode for the WAN interface.

Example:

```
Router(config)# interface gi0/0/0
Router(config-if)#
```

b) Use the **ipv6 enable** command to enable IPv6 support on the interface.

Example:

```
Router(config-if)# ipv6 enable
```

c) Use the **ip address** *ip-address subnet-mask* command to assign an IP address to the WAN interface.

Example:

```
Router(config-if)# ip address 12.0.0.1 255.255.255.0
```

d) Use the **ip dlep vtemplate** *number* **port** *udp-port* **tcp port** *tcp-port* **client ip** *ip-address* **port** *client-port* command to attach the DLEP template to the WAN interface.

Example:

```
Router(config-if)# ip dlep vtemplate 1 port 11117 tcp port 11115 client ip 12.0.0.2 port
 859
```

e) Use the **no shutdown** command to bring up the interface.

Example:

```
Router(config-if)# no shutdown
```

f) Use the **exit** command to return to global configuration mode.

Example:

```
Router(config-if)# exit
```

DLEP is configured with OSPFv3, enabling DLEP communication between the router and radio devices for adaptive routing based on link characteristics.

Configure DLEP with EIGRP

Use this task to configure Dynamic Link Exchange Protocol (DLEP) with EIGRP to enable dynamic link metric updates and improved routing decisions in mobile network environments. This task creates a VMI, binds it to a physical interface, and enables the EIGRP routing protocol.

Perform this configuration when you need to integrate DLEP with EIGRP for dynamic routing metric adjustments based on radio link conditions.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
                        Router(config)#
```

Step 2 Use the **interface vmi** *number* command to create the VMI and enter interface configuration mode.

Example:

```
Router(config)# interface vmi 1
                        Router(config-if)#
```

Step 3 Use the **ip unnumbered** *interface* command to configure the VMI to use the IP address of the physical interface.

Example:

```
Router(config-if)# ip unnumbered GigabitEthernet 0/0/0
```

Step 4 Use the **physical-interface** *interface* command to bind the physical interface to the VMI for packet flow.

Example:

```
Router(config-if)# physical-interface GigabitEthernet 0/0/0
```

Step 5 Use the **ipv6 enable** command to enable IPv6 support on the VMI.

Example:

```
Router(config-if)# ipv6 enable
```

Step 6 Use the **no ip split-horizon eigrp** *as-number* command to disable split horizon for the EIGRP process.

Example:

```
Router(config-if)# no ip split-horizon eigrp 1
```

Step 7 Use the **exit** command to exit interface configuration mode and return to global configuration mode.

Example:

```
Router(config-if)# exit
                        Router(config)#
```

Step 8 Use these steps to enable the global configuration for EIGRP:

a) Use the **router eigrp** *as-number* command to enter router configuration mode.

Example:

```
Router(config)# router eigrp 1
                                    Router(config-router)#
```

b) Use the **router-id** *ip-address* command to configure the router ID.

Example:

```
Router(config-router)# router-id 1.1.1.1
```

c) Use the **network** *ip-address wildcard-mask* command to enable all the networks that can be part of EIGRP.

Example:

```
Router(config-router)# network 10.0.0.0 0.0.0.255
```

DLEP is configured with EIGRP, and the VMI is bound to the physical interface with EIGRP routing enabled for the specified network.

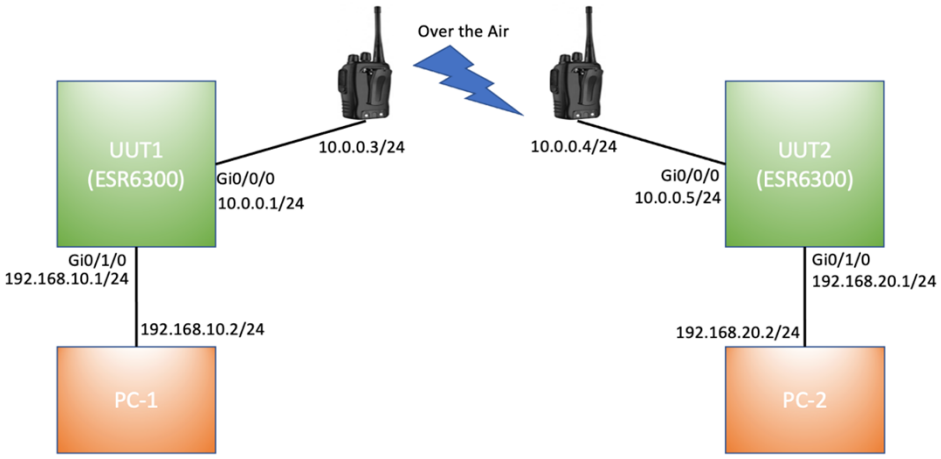
DLEP with quality of service (QoS)

Quality of Service (QoS) for DLEP needs to be configured on the Virtual-Template which is associated with the physical interface. QoS policy can be verified using the command **sh policy-map interface Virtual-Access**, as the actual data packets for DLEP will flow through the Virtual Access interfaces that will be created when DLEP neighbors come up.

Before proceeding, it is a good idea to familiarize yourself with content in the Quality of Service (QoS) Configuration Guide for IOS-XE.

DLEP QoS example

This figure shows an example topology for DLEP with QoS.



Based on above figure, the QoS policy is applied to egress of Virtual-Template attached to WAN interface Gi0/0/0.

UUT1 running configuration

```
UUT1# show running-config
Building configuration...

Current configuration : 7773 bytes
!
!
version 17.8
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
!
hostname UUT1
!
boot-start-marker
boot system bootflash:/c6300-universalk9.SSA.bin
!
ipv6 unicast-routing
!
class-map match-any CMAP_VIDEO
 match dscp 33
 match dscp 35
 match dscp 37
 match dscp 39
 match dscp af41
class-map match-any CMAP_VOICE
 match dscp 41
 match dscp 43
 match dscp 45
 match dscp 47
```

```
 match dscp 49
class-map match-any CMAP_DATA
 match dscp 9
 match dscp 11
 match dscp 13
 match dscp 15
 match dscp af11
!
policy-map Queue_Map
 class CMAP_VOICE
  bandwidth percent 40
  set dscp af11
 class CMAP_VIDEO
  bandwidth percent 50
 class CMAP_DATA
  bandwidth percent 10
  set dscp af23
!
interface Loopback1
 ip address 1.1.1.1 255.255.255.255
 ipv6 enable
 ospfv3 1 ipv4 area 0
!
interface GigabitEthernet0/0/0
 ip address 10.0.0.1 255.255.255.0
 ip dlep vtemplate 1 port 11113 tcp port 11114 client ip 10.0.0.3 port 11115
 negotiation auto
 ipv6 address 1000::1/64
 ipv6 enable
!
interface GigabitEthernet0/0/1
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0/1/0
 switchport access vlan 30
!
interface GigabitEthernet0/1/1
!
interface GigabitEthernet0/1/2
!
interface GigabitEthernet0/1/3
!
interface Virtual-Template1
 ip unnumbered GigabitEthernet0/0/0
 ipv6 enable
 service-policy output Queue_Map
!
interface Vlan1
 no ip address
!
interface Vlan30
 ip address 192.168.10.1 255.255.255.0
 ipv6 address 1010::1/64
 ipv6 enable
 ospfv3 1 ipv6 area 0
!
interface Async0/2/0
 no ip address
 encapsulation scada
!
interface vmi1
 ip unnumbered GigabitEthernet0/0/0
```

```
 ipv6 address FE80::7E31:EFF:FE85:1E78 link-local
 ipv6 enable
 ospfv3 1 ipv4 area 0
 physical-interface GigabitEthernet0/0/0
!
router ospfv3 1
 !
 router-id 1.1.1.1
 address-family ipv4 unicast
 exit-address-family
!
end
UUT1#
```

Validation for DLEP QoS

For the above example, where QoS policy is applied to Virtual-Template1, data packets are flowing through Virtual-Access2 interface which is created when DLEP neighbors came up.



Note The CLI **show policy-map interface Virtual-Template1** will not show the stats.

```
UUT1#show policy-map interface Virtual-Template 1
 Virtual-Template1
Service-policy output: Queue_Map
```

Service policy content is displayed for cloned interfaces only such as virtual access and sessions.

See the output for Virtual-Access 2:

```
UUT1#show policy-map interface Virtual-Access 2
Virtual-Access2
Service-policy output: Queue_Map

Class-map: CMAP_VOICE (match-any)
0 packets, 0 bytes
5 minute offered rate 0000 bps, drop rate 0000 bps
Match:  dscp 41
Match:  dscp 43
Match:  dscp 45
Match:  dscp 47
Match:  dscp 49
Queueing
queue limit 208 packets
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts output/bytes output) 0/0
bandwidth 40% (400000 kbps)
QoS Set
dscp af11
Marker statistics: Disabled

Class-map: CMAP_VIDEO (match-any)
0 packets, 0 bytes
5 minute offered rate 0000 bps, drop rate 0000 bps
Match:  dscp 33
Match:  dscp 35
Match:  dscp 37
Match:  dscp 39
Match:  dscp af41 (34)
Queueing
queue limit 208 packets
(queue depth/total drops/no-buffer drops) 0/0/0
```

```
(pkts output/bytes output) 0/0
bandwidth 50% (500000 kbps)

Class-map: CMAP_DATA (match-any)
0 packets, 0 bytes
5 minute offered rate 0000 bps, drop rate 0000 bps
Match:  dscp 9
Match:  dscp 11
Match:  dscp 13
Match:  dscp 15
Match:  dscp af11 (10)
Queueing
queue limit 208 packets
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts output/bytes output) 1024337/34827458
bandwidth 10% (100000 kbps)
QoS Set
dscp af23
Marker statistics: Disabled

Class-map: class-default (match-any)
0 packets, 0 bytes
5 minute offered rate 0000 bps, drop rate 0000 bps
Match: any

queue limit 208 packets
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts output/bytes output) 0/0
```

Edit the virtual template

Before you begin

Before you edit the virtual template, remove the **ip dlep vtemplate** configuration from the WAN interface.

Use this task to modify virtual template interface configurations by creating or editing the interface and applying service policies for traffic management.

Virtual template interfaces are used to create virtual interfaces dynamically. Perform this configuration when you need to manage traffic policies on these interfaces.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
Router(config)#
```

Step 2 Use the **interface Virtual-Template** *number* command to create the virtual template interface and enter interface configuration mode.

Example:

```
Router(config)# interface Virtual-Template 1
Router(config-if)#
```

Step 3 Use the **service-policy** *{input | output} policy-map-name* command to apply the policy map to the ingress or egress direction of the virtual template interface.

Example:

```
Router(config-if)# service-policy output Queue_Map
```

The virtual template interface is configured with the specified policy map applied for traffic management.

Configure DLEP on a sub-interface

Use this task to configure Dynamic Link Exchange Protocol (DLEP) on a sub-interface to enable dynamic link monitoring and management capabilities on a VLAN-tagged interface.

DLEP can also be configured on a sub-interface. This configuration is useful when you need to implement DLEP functionality on specific VLAN segments rather than on the main physical interface.

Procedure

Step 1 Use these **interface** commnds:

a) Use the **interface** *type number* command to select the WAN interface, and then use the **no shutdown** command to bring it to the active state.

Example:

```
Router(config)# interface gi0/0/0
Router(config-if)# no shutdown
```

b) Use the **interface** *type number.subinterface-number* command to create the sub-interface.

Example:

```
Router(config-if)# interface gi0/0/0.2
```

Step 2 Use the **encapsulation dot1q** *vlan-id* **native** command to add 802.1Q encapsulation over the VLAN and configure it as the native VLAN.

Example:

```
Router(config-subif)# encapsulation dot1q 2 native
```

Step 3 Use these commands to assign and enable IPv4 and IPv6 addresses:

a) Use the **ip address** *ip-address subnet-mask* command to assign an IPv4 address to the sub-interface.

Example:

```
Router(config-subif)# ip address 10.0.0.1 255.255.255.0
```

b) Use the **ipv6 enable** command to enable IPv6 support on the sub-interface.

OSPFv3 IPv4 requires IPv6 support to be enabled at the interface level.

Example:

```
Router(config-subif)# ipv6 enable
```

Step 4 Use the **interface vmi** *number* command to create the virtual multipoint interface (VMI) and enter interface configuration mode.

Example:

```
Router(config-subif)# interface vmi 1
Router(config-if)#
```

Step 5 Use the **ip unnumbered** *interface* command to configure the VMI to use the IP address of the physical interface.

Example:

```
Router(config-if)# ip unnumbered GigabitEthernet 0/0/0.2
```

Step 6 Use the **physical-interface** *interface* command to bind the physical interface to the VMI for packet flow.

Example:

```
Router(config-if)# physical-interface GigabitEthernet 0/0/0.2
```

Step 7 Use the **ipv6 enable** command to enable IPv6 support on the VMI.

OSPFv3 IPv4 requires IPv6 support to be enabled at the interface level.

Example:

```
Router(config-if)# ipv6 enable
```

a) Use the **interface** *type number.subinterface-number* command to return to the sub-interface configuration to add the DLEP template.

Example:

```
Router(config-if)# interface gi0/0/0.2
```

Step 8 Use the **ip dlep vtemplate** *number* **gtsm client** *ip-address* command to attach the DLEP template to the sub-interface.

Example:

```
Router(config-subif)# ip dlep vtemplate 1 gtsm client 10.0.0.2
```

a) Use the **exit** command to exit sub-interface configuration mode and return to global configuration mode.

Example:

```
Router(config-subif)# exit
Router(config)#
```

Step 9 Use these steps to enable the global configuration for EIGRP:

a) Use the **router eigrp** *as-number* command to enter router configuration mode.

Example:

```
Router(config)# router eigrp 1
```

b) Use the **router-id** *ip-address* command to configure the router ID.

Example:

```
Router(config-router)# router-id 1.1.1.1
```

c) Use the **network** *ip-address wildcard-mask* command to specify the network.

Example:

```
Router(config-router)# network 10.0.0.0 0.0.0.255
```

DLEP is successfully configured on the sub-interface with the associated VMI and routing protocol configuration. The sub-interface is ready to handle DLEP communications.

The following is an example of the running configuration:

```
Router# show running-config
        Building configuration...
        Current configuration : 7726 bytes
        !
        !
        version 17.8
        hostname Router
        !
        boot-start-marker
        boot system bootflash:/c6300-universalk9.SSA.bin
        boot-end-marker
        !
        ipv6 unicast-routing
        subscriber templating
        !
        license udi pid ESR-6300-CON-K9 sn FOC234304H3
        license boot level network-advantage
        !
        interface Loopback1
        ip address 1.1.1.1 255.255.255.255
        ipv6 enable
        ospfv3 1 ipv4 area 0
        !
        interface GigabitEthernet0/0/0
        no ip address
        negotiation auto
        !
        interface GigabitEthernet0/0/0.10
        encapsulation dot1Q 10
        ip address 10.0.0.1 255.255.255.0
        ip dlep vtemplate 1 port 11113 tcp port 11114 client ip 10.0.0.2 port 11115
        ipv6 enable
        !
        interface Virtual-Template1
        ip unnumbered GigabitEthernet0/0/0.10
        !
        interface vmi1
        ip unnumbered GigabitEthernet0/0/0.10
        ipv6 address FE80::7E31:EFF:FE85:1E78 link-local
        ipv6 enable
        ospfv3 1 network manet
        ospfv3 1 ipv4 area 0
        physical-interface GigabitEthernet0/0/0.10
        !
        router ospfv3 1
        !
        address-family ipv4 unicast
        exit-address-family
        !
        address-family ipv6 unicast
        exit-address-family
        !
        end
```

Remove the DLEP configuration

Use this task to remove the DLEP configuration so that you can edit the virtual template or VMI interface configurations without encountering configuration conflicts.

Before you edit or remove any configuration related to the virtual template or the VMI, you must remove the DLEP virtual template configuration attached to the WAN interface or sub-interface. If you attempt to edit the VMI or the virtual template, the following messages are displayed:

```
Router(config)# interface vmi1
  vmi1 is associated with an instance of DLEP running on GigabitEthernet0/0/0.
Please remove DLEP configuration before making any changes to the VMI configuration.
```

```
Router(config)# interface Virtual-Template1
Virtual-Template1 is associated with an instance of DLEP running on GigabitEthernet0/0/0.
Please remove DLEP configuration before making any changes to the virtual-template.
```

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
Router(config)#
```

Step 2 Use the **interface** *type number* command to enter interface configuration mode for the interface on which DLEP is configured.

Example:

```
Router(config)# interface gi0/0/0
```

Step 3 Use the **no ip dlep vtemplate** *number* command to detach the DLEP configuration.

Example:

```
Router(config-if)# no ip dlep vtemplate 1
```

Step 4 Use the **no interface vmi** *number* command to remove the VMI from the router.

Example:

```
Router(config-if)# no interface vmi 1
```

Step 5 Use the **no interface Virtual-Template** *number* command to remove the virtual template from the router.

Example:

```
Router(config-if)# no interface Virtual-Template 1
```

DLEP is removed from the router. However, the virtual access interfaces that were created when the DLEP neighbors came up continue to appear in the output of the **show ip interface brief** command until the system is rebooted.

Configure DLEP using the WebUI

Use this task to configure Dynamic Link Exchange Protocol (DLEP) on physical interfaces, sub-interfaces, and virtual templates using the web user interface (WebUI), and to monitor DLEP neighbors and clients.

DLEP configuration allows for dynamic management of network links. The WebUI provides a graphical interface for configuring DLEP on different interface types, including physical interfaces, sub-interfaces, and virtual templates.

The following images illustrate how to configure DLEP using the WebUI.

Procedure

Step 1 To configure DLEP on a physical interface, navigate to **Configuration** > **Interface** > **Ethernet**, select the interface to configure, apply the DLEP virtual template, and then click **Update & Apply to Device**.

Step 2 To configure DLEP on a sub-interface, navigate to **Configuration** > **Interface** > **Ethernet**, select the sub-interface to configure, apply the DLEP virtual template, and then click **Update & Apply to Device**.

Example:

Step 3 Use these steps to configure the virtual template:

a) Navigate to **Configuration** > **Interface** > **Logical** and select the **Virtual-Template** tab.

From this tab, you can add, edit, or delete interfaces.

Example:

b) Click **Add** to add a virtual template, or select an existing virtual template and click **Edit**.

Example:

The **Add Virtual Interface** window appears.

c) Click **Apply to Device**.

Step 4 Use these steps to configure the virtual multipoint interface (VMI):

a) Navigate to **Configuration** > **Interface** > **Logical** and select the **VMI** tab.

From this tab, you can add, edit, or delete interfaces.

Example:

b) Click **Add** to add a VMI, or select an existing VMI and click **Edit**.

Example:

The **Add VMI** window appears.

c) Click **Apply to Device**.

Step 5 To monitor DLEP, navigate to **Monitoring** > **General** > **DLEP**.

Two tabs are available for monitoring neighbors or clients. The **Neighbors** tab displays information equivalent to the **show dlep neighbors** command. The **Clients** tab displays information equivalent to the **show dlep clients** command.

Step 6 Use these steps to monitor DLEP neighbors:

a) On the **Neighbors** tab, select the interface.

Example:

b) Open the interface to view the details.

Example:

Step 7 To monitor DLEP clients, select the interface on the **Clients** tab and open it to view the details.

The **Clients** tab works in the same manner as the **Neighbors** tab.

Example:

DLEP is successfully configured on the selected interfaces, and monitoring capabilities are available through the WebUI. You can now monitor DLEP neighbors and clients through the monitoring interface.

DLEP validation commands

This section contains examples of how to verify the DLEP configuration on the router.

DLEP configuration

Command	Information

`Router# show dlep config ?`
`  GigabitEthernet  GigabitEthernet IEEE 802.3z`
 	
`  |  Output modifiers`
`  <cr>  <cr>`
	DLEP configuration is supported only on WAN or sub-interface.

```
Router# show dlep config g0/0/1
DLEP Configuration for GigabitEthernet0/0/1

DLEP Peer Description -
DLEP Version = RFC 8175
DLEP Server IP=10.0.0.1:11117
DLEPv27 TCP Port = 11118
Virtual template=2
Timers (all values are in seconds):
Missed heartbeat threshold=2, Peer Terminate ACK timeout=10
Dlepv27 Applicable configs(in seconds):
Heartbeat interval=5, Discovery interval =5, Session Ack timeout=10
Neighbor activity timeout=0, Neighbor Down ACK timeout=10
```

DLEP clients

Command	Information

`Router# show dlep clients ?`
`  GigabitEthernet  GigabitEthernet IEEE 802.3z`
 	
`  |  Output modifiers`
`  <cr>  <cr>`
	DLEP clients is supported only on WAN or sub-interface.

```
Router# show dlep clients
DLEP Clients for all interfaces:



DLEP Clients for Interface GigabitEthernet0/0/1
DLEP Server IP=10.0.0.1:11117 Sock=0 --> Local Router IP address


DLEP Client IP=10.0.0.2:859 TCP Socket fd=1 --> Directly connected Radio to the router
Peer ID=2, Virtual template=2
Description: DLEP_RadioSIM2
Peer Timers (all values in milliseconds):
  Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

Supported Metrics:
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
  Link MTU Metric : 100
  Link Latency Metric : 250 microseconds
  Link CDR RX Metric : 100000000 bps
  Link CDR TX Metric : 100000000 bps
  Link MDR RX Metric : 100000000 bps
  Link MDR TX Metric : 100000000 bps
Router#
```

DLEP neighbor

Command	Information
Router# **show dlep neighbor** ? GigabitEthernet GigabitEthernet IEEE 802.3z | Output modifiers <cr> <cr>	DLEP neighbors is supported only on WAN or sub-interface.

```
Router# show dlep neighbor
DLEP Neighbors for all interfaces:

DLEP Neighbors for Interface GigabitEthernet0/0/1
DLEP Server IP=10.0.0.1:11117 Sock=0   ---> Local Router IP address

SID=2151  MAC_Address=a453.0e94.f861
  Addresses:
  IPv4 : 16.0.0.1          ---> Mac-Address and IPv4 address of neighbor's end-point device
Supported Metrics:
  RLQ RX Metric : 100
  RLQ TX Metric : 100
  Resources Metric : 100
  MTU Metric : 1500
  Latency Metric : 250 microseconds
  CDR RX Metric : 100000000 bps
  CDR TX Metric : 100000000 bps
  MDR RX Metric : 100000000 bps
  MDR TX Metric : 100000000 bps
```

DLEP counters

Command	Information

Router# **show dlep counters ?**
  GigabitEthernet GigabitEthernet IEEE 802.3z
 	
  | Output modifiers
  <cr> <cr>
	DLEP Counters is supported only on WAN or sub-interface, which will summarize port information, counters for peer, and neighbors.

```
UUT1# show dlep counters
DLEP Counters for GigabitEthernet0/0/1

Last Clear Time =

DLEP Version = RFC 8175
DLEP Server IP=10.0.0.1:11117
DLEPv5 TCP Port = 11118

Peer Counters:
RX Peer Discovery     0      TX Peer Offer             0
 RX Peer Offer         0      TX Peer Discovery         0
 RX Peer Init          0      TX Peer Init Ack          0
 RX Peer Init Ack      1      TX Peer Init              1
 RX Heartbeat          41     TX Heartbeat              41
 RX Peer Terminate     0      TX Peer Terminate Ack     0
 RX Peer Terminate Ack 0      TX Peer Terminate         0

Neighbor Counters:
RX Neighbor Up        1      TX Neighbor Up Ack        1
 RX Metric             0
 RX Neighbor Down      0      TX Neighbor Down Ack      0
 RX Neighbor Down Ack  0      TX Neighbor Down          1

Exception Counters:
RX Invalid Message    0      RX Unknown Message        0
 Pre-Existing Neighbor 0      Neighbor Resource Error   1
 Neighbor Not Found    0      Neighbor Msg Peer Not Up  0

Timer Counters:
Peer Heartbeat Timer         41
 Peer Terminate Ack Timer     0
 Neighbor Terminate Ack Timer 0
 Neighbor Activity Timer      0
 Radio Connect Timer          5

Single Timer Wheel "Manet Infra Wheel"
  Granularity      = 250 msec
  Wheel size       = 4096
  Spoke index      = 3730
  Tick count       = 3423890
  Flags            = 0x00
  Active timers    = 1
  High water mark  = 1
  Started timers   = 171177
  Restarted timers = 2
  Cancelled timers = 5
  Expired timers   = 171169
  Long timers      = 0
  Long timer revs  = 0
  Timer suspends   = 0
```

Optional configurations for DLEP

There are set of optional commands that are available to configure for the **ip DLEP** CLI under the WAN or sub-interface. Based on those commands, you can define the set of timeout intervals between peers, neighbors, and how long to send heart-beat intervals to radio's.

Command	Purpose
ip dlep set heartbeat-threshold ? `<2-8>  Threshold of missed heartbeat messages`	Set the heartbeat-threshold, between Server and Client.
ip dlep set nbr-activity-timeout ? `<0-240>  Neighbor Activity timer duration in seconds`	Set the neighbor activity timeout.
ip dlep set nbr-down-ack-timeout ? `<1-50>  Neighbor Down ACK timer duration in seconds`	Set the neighbor acknowledgment timeout.
ip dlep set peer-description ? `LINE  Peer Description Name`	Defines the description with Peer.
ip dlep set peer-heartbeat-interval ? `<1-60>  Peer Heartbeat Interval timer duration in seconds`	Set the heartbeat interval between Server and client.
ip dlep set peer-discovery-interval ? `<1-60>  Peer Discovery Interval timer duration in seconds`	Set the peer discovery interval timer.
ip dlep set peer-init-ack-timeout ? `<1-60>  Peer Init ACK timer duration in seconds`	Set the peer with init acknowledge messages from client.
ip dlep set peer-terminate-ack-timeout ? `<1-50>  Peer Terminate ACK timer duration in seconds`	Set the peer terminate acknowledge timer between client and server.

Restrictions for DLEP IPv6 unicast

DLEP deployment restrictions

Previous releases of IOS XE offered support for IPv4 unicast traffic over an IPv4 DLEP session. IOS XE 17.12.1a provides support for IPv6 unicast over an IPv4 DLEP session.

- Multicast traffic is not supported with DLEP, but is supported with PPPOE.
- DLEP cannot be deployed with High Availability (HA) configuration.
- You must configure the VMI and Virtual-Template before attaching the Virtual-Template to a physical interface.
- The ESR6300 is connected over DLEP radio links and only 1 radio per interface (WAN port only) is supported.

- All configurations for the virtual-template need to be removed individually using the no form of the respective configuration commands, before removing the virtual-template using the no interface virtual-template command.
- Changing of configurations on the virtual-template and VMI interfaces is not supported while DLEP is enabled on the physical interface. In order to make such changes, disable DLEP by removing the DLEP configuration from the physical interface, make the changes, and re-configure DLEP on the physical interface.
- DLEP interface does not support Jumbo frames (frames > 1500 bytes in size).
- Routing of internally generated application traffic (e.g. pingv6) with source as DLEP VMI / physical interface is not supported.
- Viewing information about DLEP neighbors using the show ipv6 neighbor command is not supported.

This section provides a subset of the overall DLEP information that is found in the IP Routing Configuration Guide, Cisco IOS XE 17.x.

IPv4 and IPv6 multicast over DLEP

DLEP Multicast is now supported with IOS XE 17.13.1. See the configuration examples for IPv4 and IPv6 multicast implementation.

IPv4 multicast

Global Command Examples:

```
ip multicast-routing distributed
ip pim rp-address <rp ip addr>  [access-list for group]
```

Physical Interface Command Examples:

```
interface GigabitEthernet0/0/1
  ip address 28.28.28.1 255.255.255.0
  ip pim sparse-mode
  ip igmp version 3
  negotiation auto
  ipv6 address 1111::3/120
ipv6 enable
```

VMI Command Examples:

```
interface vmi 5
ip pim sparse-mode
```

Loopback Command Examples:

```
interface Loopback0
ip address 7.7.7.7 255.255.255.255
ip pim sparse-mode
ipv6 address 2000::2/128
ipv6 enable
```

IPv6 multicast

Global Command Examples:

```
ipv6 pim
ipv6 multicast-routing
ipv6 pim rp-address <rp ipv6 addr> [access-list for group]
```

Physical Interface Command Examples:

```
interface GigabitEthernet0/0/2
ip address 38.38.38.1 255.255.255.0
ip igmp version 3
ipv6 mld version 2
negotiation auto
ipv6 address 1111::3/120
ipv6 enable
ipv6 pim register-source loopback0
```

VMI Command Examples:

```
interface vmi 5
ip pim sparse-mode
```

Loopback Command Examples:

```
interface Loopback0
ip address 7.7.7.7 255.255.255.255
ip pim sparse-mode
ipv6 address 2000::2/128
ipv6 enable
```

Remove commands

Remove Command Examples:

```
Router(config-if)# no ip dlep vtemplate 1
Router(config-if)# no int vmi 1
Router(config-if)# no int Virtual-Template 1
```

Show commands

Show Command Examples:

```
show ip interface brief
show dlep config
show dlep clients
show dlep neighbors
```

IPv6 control plane for DLEP

IPv6 control plane for DLEP is a network capability that

- enables DLEP control plane session support on IPv6 networks
- allows routers to initiate DLEP session setup over IPv6 when both router and radio run IPv6 stack, and
- supports both manual mode and auto discovery mode for session establishment.

Configuration modes

Manual mode session setup occurs when the router is configured with an IPv6 address and transport layer port details of listening radio, enabling the router to initiate DLEP session setup.

DLEP client and neighbor clearing

DLEP client and neighbor clearing is a network management process that

- terminates active DLEP peer sessions between routers and radios using specific CLI commands
- handles session teardown through message exchanges or automatic timeout mechanisms, and
- provides granular control over individual client connections and neighbor relationships.

Command operation details

When the **clear DLEP client** command is issued on a router, the router sends a "Session Termination" message to the radio. The radio responds to this message by sending a "Session Termination Response" message. When the router receives the "Session Termination Response" message, it tears down the peer session.

When the radio is stopped gracefully, and after the **clear DLEP client** command is issued on the router, the router will still send "Session Termination" message to the radio, but it won't be received by the radio, since it is already turned off. The router will automatically tear down the peer session after the timeout (heartbeat timer expiry) is reached.

There is another clear DLEP command that can be used. The clear DLEP neighbor <interface> <session ID> CLI will clear DLEP neighbors. The session ID can be obtained from show DLEP neighbor command.

DLEP client clearing example



Note First obtain the Peer ID from the output of the **show DLEP client** CLI. Then, use that as the input to the **clear DLEP client** CLI.

```
Router#show dlep client gi0/0/1

DLEP Clients for Interface GigabitEthernet0/0/1
DLEP Local IP=15.0.0.10:55555 Sock=0


DLEP Local Radio IP=15.0.0.2:856 TCP Socket fd=1
Peer ID=20, Virtual template=2
Description: DLEP_Radio_2042
Peer Timers (all values in milliseconds):
Heartbeat=60000, Dead Interval=120000, Terminate ACK=240000
Neighbor Timers (all values in seconds):
Activity timeout=0, Neighbor Down ACK=10

Supported Metrics:
Link RLQ RX Metric : 100
Link RLQ TX Metric : 100
Link Resources Metric : 100
Link MTU Metric : 100
Link Latency Metric : 250 microseconds
Link CDR RX Metric : 100000000 bps
Link CDR TX Metric : 100000000 bps
Link MDR RX Metric : 100000000 bps
Link MDR TX Metric : 100000000 bps
Router#
```

```
Router#clear dlep client gi0/0/1 20
DLEP: Clear Client (peer) peer_id=20 from 15.0.0.10

Router#clear dlep neighbor gi0/0/1 2215
DLEP: Clear neighbor sid=2215 from 195.0.0.2
```

Troubleshoot with show commands

This series of figures illustrates what the output of various show commands mean in a sample installation.

Figure 10: Sample installation

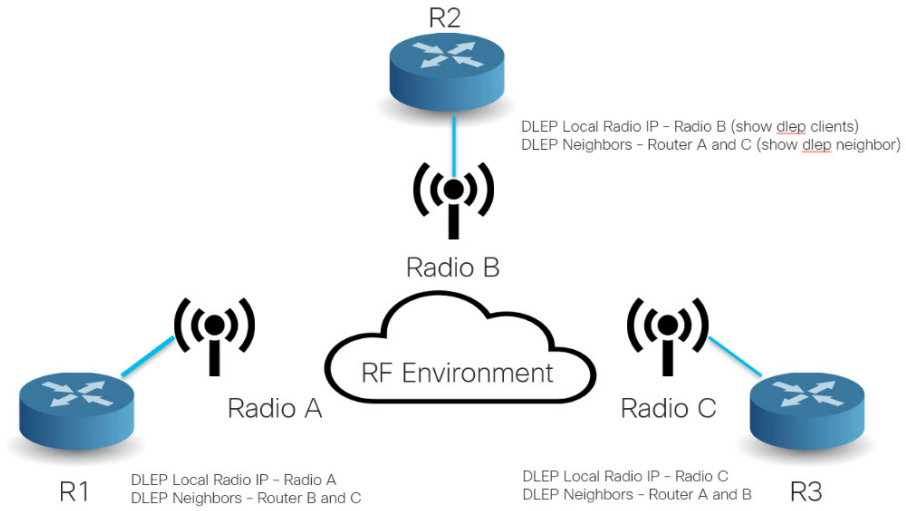


Figure 11: Show DLEP configuration



Figure 12: Show DLEP clients

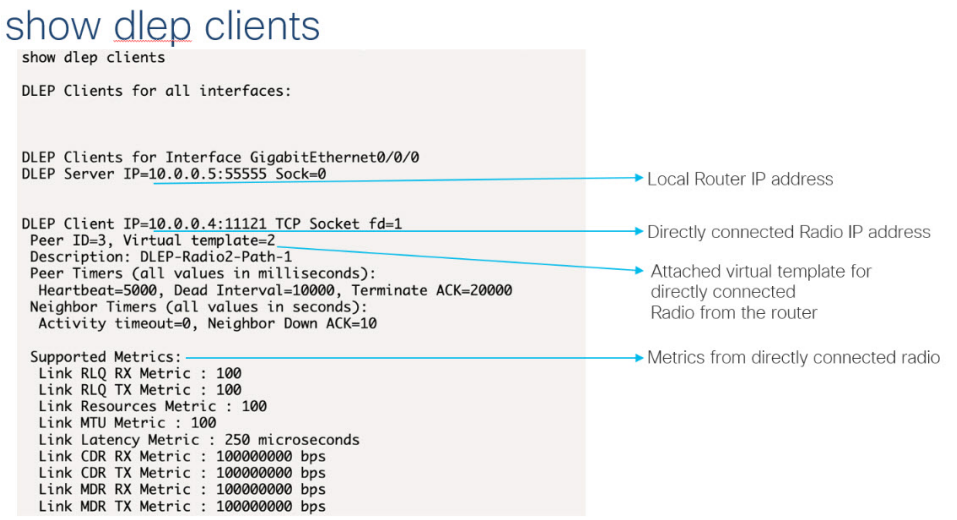


Figure 13: Show DLEP neighbors

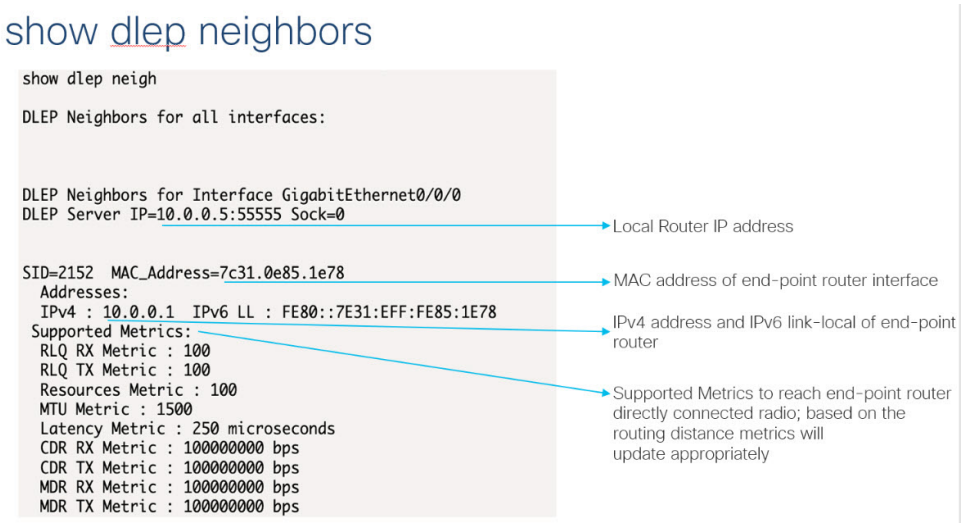
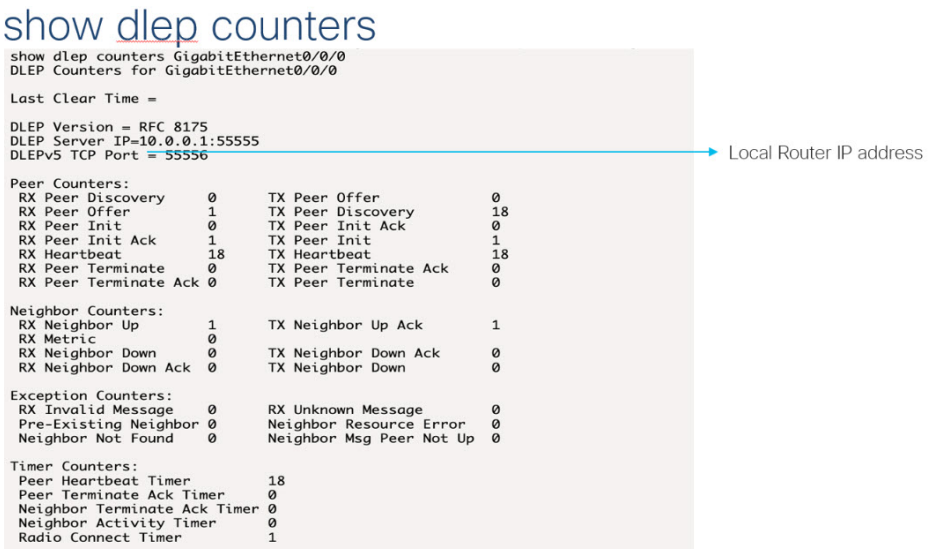


Figure 14: Show DLEP counters



Troubleshoot with debug commands

This section provides procedures to resolve DLEP client reachability issues and session timeout problems using diagnostic commands.

Scenario 1: DLEP client is not reachable

In this scenario, the router is not running in discovery mode, and the client/radio attributes have been explicitly configured.

```
Router# show dlep clients
                DLEP Clients for all interfaces:
                DLEP Clients for Interface GigabitEthernet0/0/1
                DLEP Server IP=14.0.0.3:11117 Sock=-1

Router# show dlep config
                DLEP Configuration for GigabitEthernet0/0/1
                DLEP Peer Description -
                DLEP Version = RFC 8175
                DLEP Server IP=14.0.0.3:11117
                DLEPv27 TCP Port = 11118
                Virtual template=2
                Timers (all values are in seconds):
                Missed heartbeat threshold=2, Peer Terminate ACK timeout=10
                Dlepv27 Applicable configs(in seconds):
                Heartbeat interval=60, Discovery interval =5, Session Ack timeout=10
                Neighbor activity timeout=0, Neighbor Down ACK timeout=10
                Router# show run int g0/0/1
                Building configuration...
                Current configuration : 245 bytes
                !
                interface GigabitEthernet0/0/1
                ip address 14.0.0.3 255.255.255.0
                ip dlep set peer-heartbeat-interval 60
                ip dlep vtemplate 2 port 11117 tcp port 11118 client ip 14.0.0.6 port 859
                negotiation auto
```

```
                ipv6 address 1111::1/120
                ipv6 enable
                end
```

Verify that the configuration on the radio (client) matches the configuration on the router (server) and that the router can reach the radio.

```
Router# show ip arp
                Protocol  Address          Age (min)  Hardware Addr   Type   Interface
                Internet  10.199.184.2            9   0013.5f22.0b4a  ARPA
GigabitEthernet0/0/0
                Internet  10.199.184.3            8   0018.7414.4e80  ARPA
GigabitEthernet0/0/0
                Internet  10.199.184.19           -   a453.0e94.f638  ARPA
GigabitEthernet0/0/0
                Internet  14.0.0.2                4   000c.297a.6b3d  ARPA
GigabitEthernet0/0/1
                Internet  14.0.0.3                -   a453.0e94.f639  ARPA
GigabitEthernet0/0/1
                Internet  14.0.0.6                0   Incomplete      ARPA
                Router# ping 14.0.0.6
                Type escape sequence to abort.
                Sending 5, 100-byte ICMP Echos to 14.0.0.6, timeout is 2 seconds:
                ...
                Success rate is 0 percent (0/3)
```

The ping to client 14.0.0.6 failed. A check of the radio configuration revealed that the client IP address was 14.0.0.2. Correct the client IP address and verify the fix.

```
Router# conf t
                Router(config)# int g0/0/1
                Router(config-if)# no ip dlep vtemplate 2 port 11117 tcp port 11118 client
 ip 14.0.0.6 port 859
                Router(config-if)# ip dlep set peer-heartbeat-interval 60
                Router(config-if)# ip dlep vtemplate 2 port 11117 tcp port 11118 client ip
 14.0.0.6 port 859
                Router(config-if)# ^Z
```

Scenario 2: DLEP session keeps timing out

In this scenario, the router is running in discovery mode and the DLEP session flaps.

```
Router# show dlep clients
                DLEP Clients for all interfaces:
                DLEP Clients for Interface GigabitEthernet0/0/1
                DLEP Server IP=14.0.0.3:55555 Sock=0
                DLEP Client IP=14.0.0.2:859 TCP Socket fd=1
                Peer ID=13, Virtual template=2
                Description: DLEP_RadioSIM2
```

Turn on the following debug commands to troubleshoot:

```
debug dlep server
                debug dlep timer detail
                debug dlep client error
                debug dlep client infra
                debug dlep client packet detail
                debug dlep client state
```

The debug logs indicate a connection error. An examination of the DLEP config reveals that the heartbeat on the router is set to 5 seconds, while the radio is set to 60 seconds. Change the heartbeat to 60 seconds to resolve the timeout.

```
Router# conf t
                Router(config)# int g0/0/1
```

```
Router(config-if)# no ip dlep
Router(config-if)# no ip dlep vtemplate
Router(config-if)# ip dlep set peer-heartbeat-interval 60
Router(config-if)# ip dlep vtemplate 2
Router(config-if)# ^Z
```

Additional debug commands

These commands are available for troubleshooting purposes.



Note We recommend using debug commands only when under the guidance of Cisco TAC.

DLEP

```
debug dlep server detail
debug dlep timer detail
debug dlep neighbor error
debug dlep neighbor infrastructure detail
debug dlep neighbor infrastructure error
debug dlep neighbor metrics
debug dlep neighbor state
debug dlep neighbor all
debug dlep client error
debug dlep client infrastructure
debug dlep client packet dump
debug dlep client packet detail
debug dlep client state
```

VMI

```
debug vmi bma
debug vmi packet
debug vmi error
debug vmi multicast
debug vmi neighbor
debug vmi registries
```

Virtual template

```
debug vtemplate cloning
debug vtemplate error
debug vtemplate  event
debug vtemplate subinterface
```

PPPOE

```
debug pppoe errors
debug pppoe events
debug pppoe packets
debug pppoe data
```

SSS

```
debug sss error
debug sss event
```

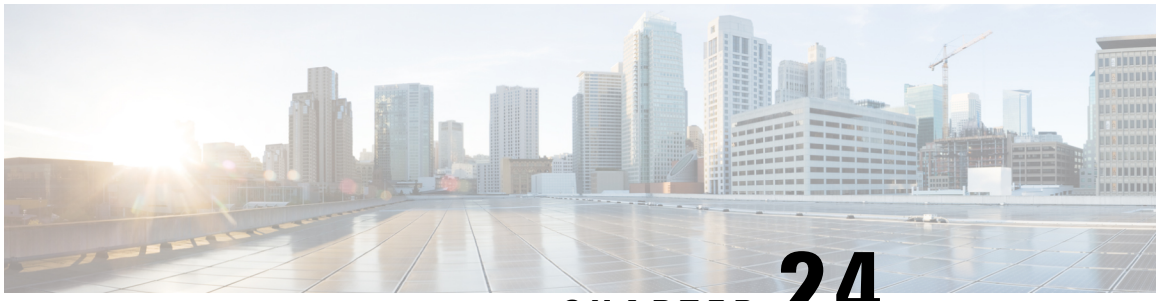
Related documentation

This reference identifies external documentation and technical standards relevant to the implementation and understanding of radio aware routing.

Additional information can be found in the following resources:

Radio Aware Routing is discussed in this Cisco white paper .

Internet Engineering Task Force (IETF) RFC 8175



CHAPTER 24

IPv6 Multicast Over PPPoE

This chapter contains the following sections:

- PPPoE in MANET, on page 255
- Requirement: feature requirements for IPV6 multicast over PPPoE, on page 258
- Enable IPv6 routing, on page 258
- Create a subscriber profile, on page 259
- Configure the license level, on page 260
- Configure PPPoE service policy, on page 260
- Configure QoS provisioning, on page 261
- Configure PPPoE service selection, on page 262
- Configure PPPoE on an ethernet interface, on page 263
- Configure a virtual template interface, on page 264
- Configure VMI interface and enable multicast support, on page 266
- Configure the loopback interface, on page 267
- Configure the OSPFv3 IPv4 address family process, on page 268
- Configure the OSPFv3 IPv6 address family process, on page 270
- Configure the IPv6 PIM bootstrap router, on page 271
- Configure an IPv6 multicast group, on page 271
- Verify virtual template interface, on page 272
- Verify PPPoE session details, on page 273
- Verify BSR election, on page 274
- IPv6 multicast routing table, on page 275
- IPv6 Multicast over PPPoE , on page 276
- Verify VMI neighbors, on page 276
- Verify OSPF neighbor, on page 278
- Sample Running Configuration, on page 278
- Debug commands, on page 283

PPPoE in MANET

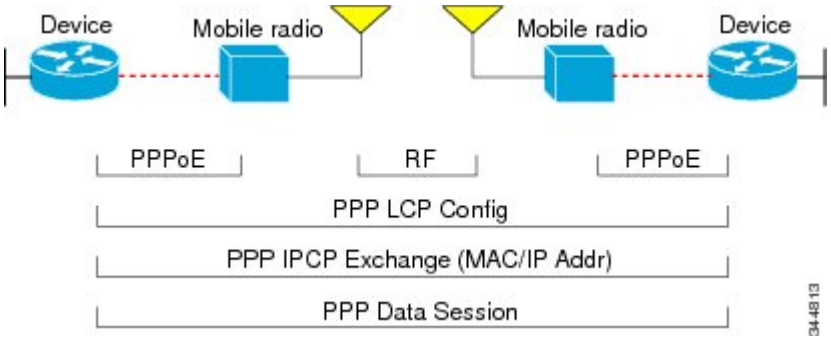
PPPoE combines Ethernet and PPP, to provide an authenticated method of assigning IP addresses to client systems.

The Cisco MANET solution employs PPPoE sessions to enable intra-nodal communications between a router and its partner radio. Each radio initiates the PPPoE session as soon as the radio establishes a radio link to another radio. After the PPPoE sessions are active, a PPP session is established end-to-end (router-to-router).

This is duplicated each time a radio establishes a new radio link. The Virtual Multipoint Interface (VMI) on the device can aggregate multiple PPPoE sessions and multiplex them to look like a single interface to the routing processes. Underneath the VMI are virtual access interfaces that are associated with each of the PPP and PPPoE connections.

A PPPoE session is established between a device and a radio on behalf of every other device and radio neighbor located in the MANET. These Layer 2 sessions are the means by which radio network status gets reported to the Layer 3 processes in the device. The following figure shows the PPPoE session exchange between mobile devices and directional radios in a MANET network.

Figure 15: PPPoE Session Exchange Between Mobile Devices and Directional Radios



Link Control Protocol (LCP) forms part of the Point-to-Point Protocol (PPP), within the family of Internet protocols. The routers exchange IP Control Protocol (IPCP) messages to negotiate options specific to the protocol.

VMI Interface

The VMI on the device can aggregate all of the per-neighbor PPPoE sessions from the radio Ethernet connection. The VMI maps the sessions to appear to Layer 3 routing protocols and applications as a single point-to-multipoint, multiaccess, broadcast-capable network. However, the VMI preserves the integrity of the PPPoE sessions on the radio side so that each point-to-point connection can have its own quality of service (QoS) queue.

The VMI also relays the link-quality metric and neighbor up/down signaling from the radio to the routing protocols. The VMI signals are used by the Enhanced Interior Gateway Routing Protocol (EIGRP) for IPv4 and IPv6 neighbors and the Open Shortest Path First version 3 (OSPFv3) for IPv6 neighbors.

The VMI can operate in two modes: bypass or aggregate.

Bypass Mode

This is the recommended mode for PPPOE in a MANET network.

In bypass mode, the virtual-access interfaces are directly exposed to applications running above L2. In bypass mode, you must still define a VMI because VMI continues to manage presentation of cross-layer signals, such as, neighbor up, neighbor down, and metrics. However, applications will still be aware of the actual underlying virtual-access interfaces and send packets to them directly.

If you are running multicast applications that require virtual-access interfaces to be exposed to applications above L2 directly, you can configure VMI to operate in bypass mode. Most multicast applications require that the virtual-access interfaces be exposed directly to routing protocols for the multicast Reverse Path Forwarding (RPF) to operate as expected .

Aggregate Mode

In this mode, all the virtual-access interfaces created by PPPoE sessions are aggregated logically under the configured VMI. VMI on the router can aggregate multiple PPPoE sessions and multiplex them to look like a single interface to the routing processes. Applications above Layer 2 (L2), such as Enhanced Interior Gateway Routing Protocol (EIGRP) and OSPFv3, should be defined only on VMI. Underneath VMI are virtual access interfaces that are associated with each of the PPP/PPPoE connections. Packets sent to VMI are forwarded to the correct virtual-access interface(s).

Aggregate mode VMIs operate in Non-Broadcast Multiple Access (NBMA) mode. Multicast traffic is forwarded only to the NBMA neighbors where a listener for that group is present. This is the preferred mode when operating in PIM sparse mode.

Virtual Access Interface

The Virtual-Access interfaces are logically “underneath” the VMI interface. Each Virtual-Access interface represents a “destination” which is either a routing next-hop, or a multicast group. At the bottom of the interface hierarchy is the actual physical interface connecting the router and radio. The Virtual-Access interface funnels the traffic to the physical interface for transmission to the radio device.

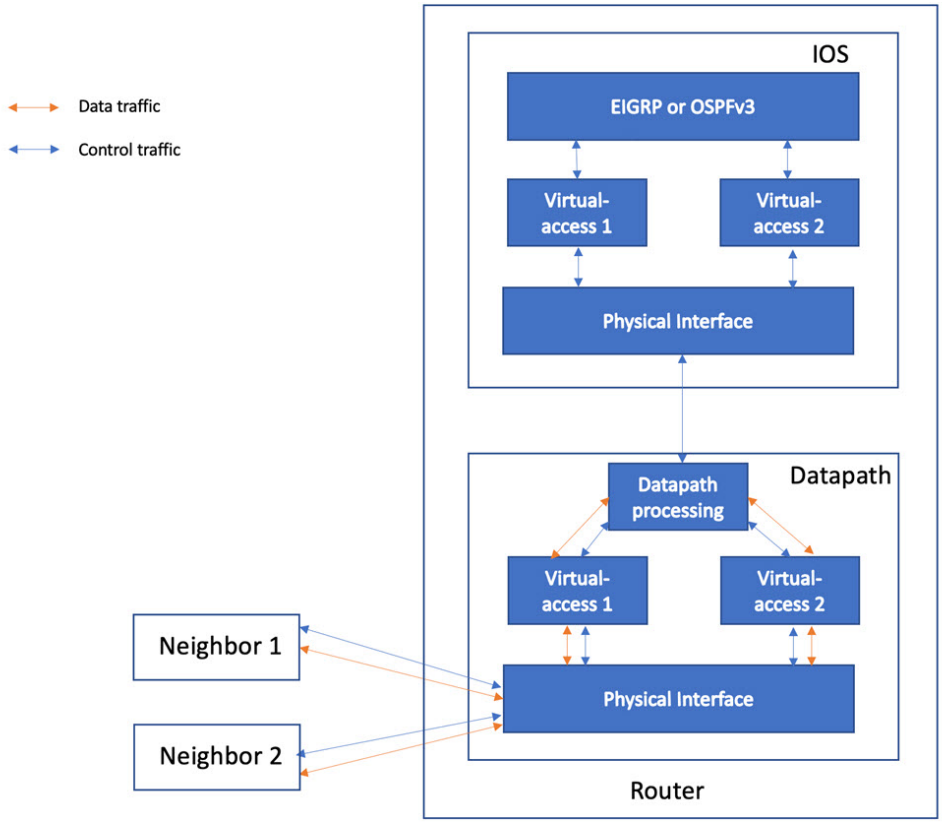
PPPOE Packet Flow

The Packet Flow diagram below illustrates the packet flow for both control and data packets over a PPPOE session when the VMI interface is in bypass mode.

All control traffic is sent to by the datapath to IOS where it is handed over to the appropriate protocol to be processed. The incoming interface for this traffic is the virtual-access interface associated with the neighbor which is the source of the traffic. In this case, Virtual-access1 corresponds to Neighbor 1 and Virtual-Access 2 corresponds to Neighbor 2.

All data traffic is processed by the datapath and does not typically get sent to IOS.

Figure 16: Packet Flow



Requirement: feature requirements for IPV6 multicast over PPPoE

For IPV6 multicast over PPPoE to function properly, these must be configured:

- PPPoE (Virtual-template, VMI and physical interface)
- IPV6 unicast and multicast routing
- IPv6 PIM BSR
- IPv6 MLD

This feature requires the Network Advantage License.

Enable IPv6 routing

Use this task to enable IPv6 unicast and multicast routing on the device.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ipv6 unicast-routing** command to enable IPv6 unicast routing.

Example:

```
Router(config)# ipv6 unicast-routing
```

Step 3 Use the **ipv6 multicast-routing** command to enable multicast routing on all IPv6-enabled interfaces, and to enable multicast forwarding for PIM and MLD on all enabled interfaces of the device.

Example:

```
Router(config)# ipv6 multicast-routing
```

IPv6 unicast and multicast routing are enabled on the device.

Create a subscriber profile

Create a subscriber profile to enable the Subscriber Service Switch Manager to determine appropriate services for each subscriber using a combination of policy and service lookup models.

The Subscriber Profile Support feature is functionality for the Subscriber Service Switch architecture, a Cisco IOS subsystem that connects subscribers to network access services at Layer 2. This functionality affects how the Subscriber Service Switch Manager determines a service for each subscriber with a combination of a policy and a service lookup model.

Note Configuring a subscriber profile for PPPoE service selection is required for VMI to function properly.

Procedure

Step 1 **configure terminal**

Example:

Router# **configure terminal**

Enters global configuration mode.

Step 2 **subscriber templating**

Example:

Router(config)#**subscriber templating**

Configure subscriber templating.

Step 3 subscriber authorization enable

Example:

Router(config)#**subscriber authorization enable**

Enable Subscriber Service Switch type authorization.

The subscriber profile is created and subscriber templating is configured with authorization enabled for the Subscriber Service Switch architecture.

Configure the license level

Use this task to configure the license level on a router to enable specific features and capabilities. The license level determines which features are available on the device.

License levels control access to router features and must be configured appropriately for your deployment requirements. Different license levels provide access to different feature sets.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **license boot level** *level* command to configure the license level.

Note
The license level must be configured as `network-advantage` for IPv6 multicast routing.

Example:

```
Router(config)# license boot level network-advantage
```

The license level is configured on the router. The specified license level becomes active after the device restarts.

Configure PPPoE service policy

Configure a PPPoE service policy to enable class-based weighted fair queueing (CBWFQ) for network traffic management.

A service policy specifies class-based weighted fair queueing (CBWFQ). The class policies that make up the policy map are then applied to packets that satisfy the class map match criteria for the class.

Procedure

Step 1 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 2 **policy-map type service** *name*

Example:

```
Router(config)#policy-map type service pppoe_rar
```

Configure policy map type.

Step 3 **pppoe service** *name*

Example:

```
Router(config-service-policymap)# pppoe service manet_radio
```

Configure service policy map.

Note
Enter the PPPoE service policy name as manet_radio.

The PPPoE service policy is configured and ready to apply class-based weighted fair queueing to network traffic.

Configure QoS provisioning

Configure QoS provisioning to establish traffic policing policies that control bandwidth usage and define actions for traffic that conforms to or exceeds specified rate limits.

QoS provisioning allows you to manage network traffic by implementing policies that define how traffic should be handled based on specific criteria. This configuration is essential for maintaining network performance and ensuring proper resource allocation.

Procedure

Step 1 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 2 **policy-map** *name*

Example:

Router(config)#**policy-map rar_policer**

Create a policy map name.

Step 3 **class** *criteria_name*

Example:

Router(config-pmap)# **class class-default**

Configure policy criteria as system default class.

Step 4 **police** *target_rate* **conform-action** *action* **exceed-action** *action*

Example:

Router(config-pmap-c)# **police 10000000 conform-action transmit exceed-action drop**

Configure QoS policy for the Virtual Template Interface.

The QoS policy map is created and configured with policing parameters that define traffic rate limits and specify actions for conforming and exceeding traffic.

Configure PPPoE service selection

The PPPoE Service Selection feature enables service providers to offer a variety of services and to charge customers according to the service chosen.

The PPPoE Service Selection feature uses service tags to enable a PPP over Ethernet (PPPoE) server to offer PPPoE clients a selection of services during call setup. The customer chooses one of the services offered, and the service is provided when the PPPoE session becomes active.

Procedure

Step 1 **configure terminal**

Example:

Router# **configure terminal**

Enters global configuration mode.

Step 2 **bba-group pppoe** *name*

Example:

Router(config)# **bba-group pppoe rar_group_1**

Configure PPPoE global group.

Step 3 **virtual-template** *number*

Example:

Router(config-bba-group)# **virtual-template 1**

Attach the Virtual Template to the PPPoE bba group.

Step 4 **service profile** *name*

Example:

Router(config-bba-group)#**service profile pppoe_rar**

Attach the service policy name .

PPPoE Service Selection is configured and the server can offer clients a selection of services during call setup.

Configure PPPoE on an ethernet interface

This task enables PPPoE sessions on an Ethernet interface, allowing the router to establish Point-to-Point Protocol connections over Ethernet for broadband connectivity scenarios.

PPPoE is commonly used in broadband deployments where authentication and session management are required. This configuration enables the interface to handle PPPoE sessions for client connections.

Before you begin

Follow these steps to configure PPPoE on an Ethernet interface:

Procedure

Step 1 **configure terminal**

Example:

Router# **configure terminal**

Enters global configuration mode.

Step 2 **interface** *[type slot/port]*

Example:

Router(config)# **interface GigabitEthernet0/0/0**

Specifies an interface type and enters interface configuration mode.

Step 3 **ipv6 enable**

Example:

Router (config-if)# **ipv6 enable**

Enable IPv6 support under VMI interface. OSPFv3 IPv4 needs to have IPv6 support enabled on the interface level.

Step 4 **pppoe enable group** *group_name*

Example:

Router(config-if)#**pppoe enable group rar_group_1**

Enables PPPoE sessions on the interface or sub interface.

PPPoE is successfully configured on the Ethernet interface and the interface is ready to handle PPPoE sessions for the specified group.

Configure a virtual template interface

Configure a virtual template interface to enable dynamic cloning of configurations for virtual access interfaces created for VMI PPPoE connections.

You use the virtual template interface to dynamically clone configurations for each virtual access interface created for a virtual multipoint interface (VMI) neighbor. You can configure multiple virtual template interfaces for your VMI PPP over Ethernet (PPPoE) connections. The selection of which virtual template to use is predicated on the service name sent by the radio during PPPoE session establishment.

Procedure

Step 1 **configure terminal**

Example:

Router# **configure terminal**

Enters global configuration mode.

Step 2 **interface Virtual-Template** *number*

Example:

Router(config)# **interface Virtual-Template 1**

Creates a Virtual Template Interface for configuration and dynamic application to Virtual Access Interfaces.

Step 3 **MTU** *size*

Example:

Router(config-if)# **MTU 1484**

Sets the MTU size.

Step 4 **IP unnumbered** *interface_type interface_number*

Example:

Router(config-if)#**IP unnumbered VMI 1**

Enable IP processing without an explicit address. Specifies the Virtual Template Interface to use the VMI interface IP address.

Step 5 **no IP redirects**

Example:

Router(config-if)#**no IP redirects**

Disables ICMP redirect messages.

Step 6 **IP tcp adjust-mss** *size*

Example:

Router(config-if)#**IP tcp adjust-mss 1444**

Adjust the mss of transit packets.

Step 7 **load-interval** *value*

Example:

Router(config-if)#**load-interval 30**

Load interval delay.

Step 8 **no peer default IP address**

Example:

Router(config-if)#**no peer default IP address**

Disables peer default IP address.

Step 9 **ipv6 enable**

Example:

Router(config-if)#**ipv6 enable**

Enable IPv6 support under Virtual Template Interface.

Step 10 **ipv6 MTU** *size*

Example:

Router(config-if)#**ipv6 MTU 1484**

Sets the IPv6 MTU size.

Step 11 **ospfv3 1 network** *type*

Example:

Router(config-if)#**ospfv3 1 network MANET**

Configure the Virtual Template Interface as MANET OSPF interface type.

Step 12 **ospfv3 1 hello-interval** *value*

Example:

Router(config-if)#**ospfv3 1 hello-interval 10**

Configure the hello interval value for OSPFv3.

Step 13 **ospfv3 1 ipv4 area** *id*

Example:

Router(config-if)#**ospfv3 1 ipv4 area 0**

Enable Virtual Template Interface to participate in OSPFv3 IPv4 routing.

Step 14 **ospfv3 1 ipv6 area** *id*

Example:

Router(config-if)#**ospfv3 1 ipv6 area 0**

Enable Virtual Template Interface to participate in OSPFv3 IPv6 routing.

Step 15 **service-policy input** *policy_map_name*

Example:

Router(config-if)#**service-policy input rar_policer**

Attach the created policy map name to the Virtual Template Interface.

The virtual template interface is configured and ready to dynamically clone configurations for virtual access interfaces created for VMI neighbors during PPPoE session establishment.

Configure VMI interface and enable multicast support

VMI interfaces provide packet mapping services and multicast support for router-to-radio communications by mapping outgoing packets to appropriate Point-to-Point Protocol over Ethernet (PPPoE) sessions based on the next-hop forwarding address.

Used in router-to-radio communications, the Virtual Multipoint Interface (VMI) interface provides services that map outgoing packets to the appropriate Point-to-Point Protocol over Ethernet (PPPoE) sessions based on the next-hop forwarding address for that packet. The VMI interface also provides a broadcast service that emulates a set of point-to-point connections as a point-to-multipoint interface with broadcast ability. When a packet with a multicast address is forwarded through the VMI interface, VMI replicates the packet and unicasts it to each of its neighbors.

Note VMI will map outgoing packets to the appropriate PPPoE sessions. It will use the next-hop forwarding address from each outgoing packet to perform this mapping.

Note VMI is required to have IP addresses assigned for VMI to work even though it will be shown as down/down while in bypass mode. The IPv4 address configured will not be advertised or used. Instead, the IPv4 address on the virtual-template will be used.

Procedure

Step 1 **configure terminal**

Example:

Router# **configure terminal**

Enters global configuration mode.

Step 2 **interface VMI** *number*

Example:

Router(config)# **interface VMI 1**

Creates a VMI and enters interface configuration mode.

Step 3 **ipv6 enable**

Example:

Router(config-if)#**ipv6 enable**

Enable IPv6 support under VMI interface. OSPFv3 IPv4 needs to have IPv6 support enabled on the interface level.

Step 4 **IP address** *address mask*

Example:

Router(config-if)#**IP address 71.0.0.1 255.255.255.0**

Configures IPv4 address under VMI interface.

Step 5 **ipv6 address** *address/prefix*

Example:

Router(config-if)#**ipv6 address 71::71/64**

Configures IPv6 address under VMI interface.

Step 6 **physical-interface** *interface*

Example:

Router(config-if)#**physical-interface gigabitEthernet {0/0/0 or 0/0/1 or 0/0/0.1}**

Binds the physical interface (the interface connected to the radio client) to the VMI interface, for packet flow.

Step 7 **mode** *name*

Example:

Router(config-if)#**mode bypass**

Do not aggregate Virtual Access Interfaces under VMI. (Must be configured as bypass).

The VMI interface is configured with IPv4 and IPv6 addresses, bound to a physical interface, and set to bypass mode for multicast support in router-to-radio communications.

Configure the loopback interface

Use this task to configure a loopback interface with IPv4 and IPv6 addresses and to enable OSPFv3 routing participation. A loopback interface provides a stable interface for router identification and routing protocols that remains active regardless of the physical interface status.

Loopback interfaces are virtual interfaces that provide a stable endpoint for router communications and routing protocol operations. They are commonly used for router identification and management access.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **interface loopback** *number* command to create the loopback interface and enter interface configuration mode.

Example:

```
Router(config)# interface loopback 1
```

Step 3 Use the **ip address** *ip-address subnet-mask* command to configure the IPv4 address for the loopback interface.

Example:

```
Router(config-if)# ip address 1.1.1.1 255.255.255.255
```

Step 4 Use the **ipv6 address** *ipv6-address/prefix-length* command to configure the IPv6 address for the loopback interface.

Example:

```
Router(config-if)# ipv6 address 11::11/128
```

Step 5 Use the **ipv6 enable** command to enable IPv6 support on the loopback interface.

Example:

```
Router(config-if)# ipv6 enable
```

Step 6 Use the **ospfv3** *process-id* **ipv4 area** *area-id* command to enable the loopback interface to participate in OSPFv3 routing for the IPv4 address family.

Example:

```
Router(config-if)# ospfv3 1 ipv4 area 0
```

Step 7 Use the **ospfv3** *process-id* **ipv6 area** *area-id* command to enable the loopback interface to participate in OSPFv3 routing for the IPv6 address family.

Example:

```
Router(config-if)# ospfv3 1 ipv6 area 0
```

The loopback interface is configured with both IPv4 and IPv6 addresses and is enabled to participate in OSPFv3 routing for both address families.

Configure the OSPFv3 IPv4 address family process

Configure the OSPFv3 IPv4 address family process to enable support for both IPv4 and IPv6 unicast traffic on your network devices.

The OSPFv3 address families feature enables both IPv4 and IPv6 unicast traffic to be supported. With this feature, you may have two device processes per interface, but only one process per AF. If the IPv4 AF is used, an IPv4 address must first be configured on the interface, but IPv6 must be enabled on the interface. A single IPv4 or IPv6 OSPFv3 process running multiple instances on the same interface is not supported.

Procedure

Step 1 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 2 **router ospfv3** *process_id*

Example:

```
Router(config)# router ospfv3 1
```

Global configuration for OSPFv3.

Step 3 **router-ID** *ipv4_address*

Example:

```
Router(config-router)#router-id 101.101.101.101
```

Router ID for OSPFv3.

Step 4 **address-family ipv4 unicast**

Example:

```
Router(config-router)#address-family ipv4 unicast
```

Adding address family for IPv4 unicast routing under global OSPFv3 configuration.

Step 5 **redistribute connected metric** *value* **metric-type** *type*

Example:

```
Router(config-router-af)#redistribute connected metric 1 metric-type
```

Redistribute metrics from external routing protocol.

Step 6 **log-adjacency-changes**

Example:

```
Router(config-if)#log-adjacency-changes
```

Logs all state changes.

The OSPFv3 IPv4 address family process is configured and ready to support both IPv4 and IPv6 unicast traffic on your network interfaces.

Configure the OSPFv3 IPv6 address family process

Configure OSPFv3 to support both IPv4 and IPv6 unicast traffic using the address families feature, enabling dual-stack routing capabilities.

The OSPFv3 address families feature enables both IPv4 and IPv6 unicast traffic to be supported. With this feature, you may have two device processes per interface, but only one process per AF. If the IPv4 AF is used, an IPv4 address must first be configured on the interface, but IPv6 must be enabled on the interface. A single IPv4 or IPv6 OSPFv3 process running multiple instances on the same interface is not supported.

Procedure

Step 1 **configure terminal**

Example:

Router# **`configure terminal`**

Enters global configuration mode.

Step 2 **router ospfv3** *process_id*

Example:

Router(config)# **`router ospfv3 1`**

Global configuration for OSPFv3.

Step 3 **router-ID** *ipv4_address*

Example:

Router(config-router)#**`router-ID 101.101.101.101`**

Router ID for OSPFv3.

Step 4 **address-family ipv6 unicast**

Example:

Router(config-router)#**`address-family ipv6 unicast`**

Adding address family for IPv6 unicast routing under global OSPFv3 configuration.

Step 5 **redistribute connected metric** *value* **metric-type** *type*

Example:

Router(config-router-AF)#**`redistribute connected metric 1 metric-type`**

Redistribute metrics from external routing protocol.

Step 6 **log-adjacency-changes**

Example:

Router(config-if)#**`log-adjacency-changes`**

Logs all state changes.

The OSPFv3 IPv6 address family process is configured and ready to support IPv6 unicast routing with the specified router ID and redistribution settings.

Configure the IPv6 PIM bootstrap router

Use this task to configure the bootstrap router (BSR) protocol for PIM-SM to provide a dynamic mechanism for distributing group-to-rendezvous point (RP) mapping information throughout an IPv6 domain. Configuring IPv6 BSR enables automatic distribution of RP mapping information and provides failover capabilities when an RP becomes unreachable.

The BSR protocol for PIM-SM provides a dynamic, adaptive mechanism to distribute group-to-RP mapping information rapidly throughout a domain. With the IPv6 BSR feature, if an RP becomes unreachable, the device detects it and modifies the mapping tables so that the unreachable RP is no longer used. The new tables are then rapidly distributed throughout the domain.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **ipv6 pim bsr candidate bsr** *ipv6-address* command to configure the device as a candidate BSR.

Use the IPv6 address of the loopback interface configured earlier.

Example:

```
Router(config)# ipv6 pim bsr candidate bsr 11::11
```

Step 3 Use the **ipv6 pim bsr candidate rp** *ipv6-address* command to configure the device as a candidate RP.

Use the IPv6 address of the loopback interface configured earlier.

Example:

```
Router(config)# ipv6 pim bsr candidate rp 11::11
```

The device is configured as both a candidate BSR and a candidate RP, enabling automatic distribution of RP mapping information throughout the IPv6 PIM domain with failover capabilities.

Configure an IPv6 multicast group

Use this task to configure an IPv6 multicast group on an interface to enable receivers to join a particular data stream, regardless of their physical or geographical location.

An IPv6 multicast group is an arbitrary group of receivers that want to receive a particular data stream. This group has no physical or geographical boundaries. Receivers can be located anywhere on the Internet or in any private network.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
Router# configure terminal
```

Step 2 Use the **interface** *interface-name* command to enter interface configuration mode for the specified interface.

Example:

```
Router(config)# interface Vlan 10
```

Step 3 Use the **ipv6 mld join-group** *multicast-group-address* command to configure the IPv6 MLD join group on the interface.

Example:

```
Router(config-if)# ipv6 mld join-group FF06:6::1
```

The IPv6 multicast group is configured on the interface, allowing the device to join the specified multicast group and receive multicast traffic.

Verify virtual template interface

This section shows examples of command output to verify your setup.

```
Router# show vtemplate
Virtual access subinterface creation is globally enabled

         Active     Active    Subint  Interface
       Interface Subinterface Capable Type
       --------- ------------ ------- ---------
Vt1            0            1   Yes   Serial

Usage Summary
                              Interface   Subinterface
                              ---------   ------------
Current Serial  in use                2             1
Current Serial  free                  0             1
Current Ether   in use                0             0
Current Ether   free                  0             0
Current Tunnel  in use                0             0
Current Tunnel  free                  0             0
Current VPN  in use                   0             0
Current VPN  free                     0             0
Total                                 2             2

Cumulative created                    3            27
Cumulative freed                      0            26
```

```
Base virtual access interfaces: 2
Total create or clone requests: 3
Cancelled create or clone requests: 0
Cumulative create request waiting for sso resources: 0
Current request queue size: 0
Current free pending: 0
Current recycle pending: 0
Current ordered recycle pending: 0

Maximum request duration: 32 msec
Average request duration: 24 msec
Last request duration: 32 msec

Maximum processing duration: 32 msec
Average processing duration: 24 msec
Last processing duration: 32 msec
Router#
```

Verify PPPoE session details

This section shows examples of command output to verify your setup.

PPPoE session status verification

```
Router-1#show pppoe session
     1 session  in LOCALLY_TERMINATED (PTA) State
     1 session  total

Uniq ID  PPPoE  RemMAC          Port       VT  VA    StateSID  LocMAC       VA-st      Type

   2     1     000c.296f.c985  Gi0/0/0    1     Vi1.1    PTA    7c31.0e85.1e78
 UP
Router#
```

```
Router-2#show pppoe session
     1 session  in LOCALLY_TERMINATED (PTA) State
     1 session  total

Uniq ID  PPPoE  RemMAC          Port       VT  VA    StateSID  LocMAC       VA-st      Type

   2     1     000c.29a1.ae42  Gi0/0/0    1     Vi1.1    PTA    d478.9b5d.0200
 UP
Router#
```

PPPoE session packet statistics

```
Router-1# show pppoe session packets all
Total PPPoE sessions 1

session id: 1
local MAC address: 7c31.0e85.1e78, remote MAC address: 000c.296f.c985
virtual access interface: Vi1.1, outgoing interface: Gi0/0/0
    67 packets sent, 67 received
    6488 bytes sent, 5908 received

PPPoE Flow Control Stats
 Local Credits: 1953   Peer Credits: 65535   Local Scaling Value 65534 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 38     PADG Timer index: 89
```

```
 PADG last rcvd Seq Num: 0
 PADG last nonzero Seq Num: 89
 PADG last nonzero rcvd amount: 0
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 36  rcvd: 0
 PADC xmit: 127926272  rcvd: 36
 In-band credit pkt xmit: 0 rcvd: 0
 Last credit packet snapshot
  PADG xmit: seq_num = 38, fcn = 65535, bcn = 0
  PADC rcvd: seq_num = 38, fcn = 1953, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 61
  PADC xmit: seq_num = 61, fcn = 1952, bcn = 0
  In-band credit pkt xmit: fcn = 0, bcn = 0
  In-band credit pkt rcvd: fcn = 0, bcn = 0
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 0
Router-1#

Router-2# show pppoe session packets all
Total PPPoE sessions 1

session id: 1
local MAC address: d478.9b5d.0200, remote MAC address: 000c.29a1.ae42
virtual access interface: Vi1.1, outgoing interface: Gi0/0/0
    78 packets sent, 75 received
    7408 bytes sent, 6642 received

PPPoE Flow Control Stats
 Local Credits: 1950   Peer Credits: 65535   Local Scaling Value 65534 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 42     PADG Timer index: 96
 PADG last rcvd Seq Num: 0
 PADG last nonzero Seq Num: 96
 PADG last nonzero rcvd amount: 0
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 40  rcvd: 0
 PADC xmit: 127795200  rcvd: 40
 In-band credit pkt xmit: 0 rcvd: 0
 Last credit packet snapshot
  PADG xmit: seq_num = 42, fcn = 65535, bcn = 0
  PADC rcvd: seq_num = 42, fcn = 1953, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 66
  PADC xmit: seq_num = 63, fcn = 1950, bcn = 0
  In-band credit pkt xmit: fcn = 0, bcn = 0
  In-band credit pkt rcvd: fcn = 0, bcn = 0
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 0
Router-2#
```

Verify BSR election

This section shows examples of command output to verify your setup.

```
Router-1#show ipv6 pim bsr election
PIMv2 BSR information

BSR Election Information
  Scope Range List: ff00::/8
  This system is the Bootstrap Router (BSR)
    BSR Address: 61::61
```

```
    Uptime: 00:05:20, BSR Priority: 0, Hash mask length: 126
    RPF: FE80::7E31:EFF:FE85:1E78,Loopback0
    BS Timer: 00:00:41
  This system is candidate BSR
      Candidate BSR address: 61::61, priority: 0, hash mask length: 126
Router-1#

Router-2#show ipv6 pim bsr election
PIMv2 BSR information

BSR Election Information
  Scope Range List: ff00::/8
    BSR Address: 61::61
    Uptime: 00:01:22, BSR Priority: 0, Hash mask length: 126
    RPF: FE80::7E31:EFF:FE85:1E78,Virtual-Access1.1
    BS Timer: 00:01:47
  This system is candidate BSR
      Candidate BSR address: 41::41, priority: 0, hash mask length: 126
Router-2#
```

IPv6 multicast routing table

This section shows examples of command output to verify your setup.

Note "*" in the source address indicates that it is a mld join request entries.

```
Router-1#show ipv6 mroute
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(A000::2, FF06:6::1), 00:01:45/00:01:44, flags: SFT
  Incoming interface: Vlan30
  RPF nbr: A000::2
  Immediate Outgoing interface list:
    Virtual-Access1.1, Forward, 00:01:45/00:02:48

Router-1#

Router-2# show ipv6 mroute
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
```

```
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF06:6::1), 00:04:33/00:03:29, RP 41::41, flags: SCL
  Incoming interface: Tunnel2
  RPF nbr: 41::41
  Immediate Outgoing interface list:
    Vlan30, Forward, 00:04:33/00:03:29

(A000::2, FF06:6::1), 00:01:41/00:02:43, flags: ST
  Incoming interface: Virtual-Access1.1
  RPF nbr: FE80::7E31:EFF:FE85:1E78
  Immediate Outgoing interface list:
    Vlan30, Forward, 00:01:41/00:02:49
Router-2
```

IPv6 Multicast over PPPoE

This new feature applies to the ESR6300 Router.

PPPoE is a session/connection-oriented protocol, which extends the point-to-point radio frequency (RF) link from an external radio to an IOS router. Router communication with the radio is represented by virtual access interface (connectivity to a radio neighbor).

VMI operates in the Bypass mode where each Virtual Access Interface (VAI) represents a radio neighbor. The VMI layer re-directs unicast routing protocol traffic to the appropriate P2P link (Virtual-Access interface) and replicates any Multicast traffic that needs to flow.

For IPV6 multicast over PPPoE to function properly, the following must be configured:

- PPPoE (Virtual-template, VMI and physical interface)
- IPV6 unicast and multicast routing
- IPv6 PIM BSR
- IPv6 MLD

Note This feature requires the Network Advantage License.

For additional information, see the IPv6 Multicast over PPPoE chapter.

Verify VMI neighbors

The Multicast for Virtual Multipoint Interfaces feature enables multicast support for RFC 5578-compliant Radio-Aware Routing (RAR). Multicast is defined as a network group membership spanning the entire network. The virtual multipoint interface (VMI) operates in aggregate mode, which means that all virtual access interfaces created by PPP over Ethernet (PPPoE) sessions are aggregated logically under the configured VMI. Packets sent to the VMI are forwarded to the correct virtual access interface. When a VMI operates in aggregate

mode, the interfaces operate in nonbroadcast multiple access (NBMA) mode. Multicast traffic is forwarded only the he NBMA neighbors where a listener for that group is present.

```
Router-1#show vmi neighbor detail
              1 vmi1 Neighbors


vmi1   IPV6 Address=FE80::D678:9BFF:FE5D:200
       IPV6 Global Addr=::
       IPV4 Address=81.0.0.1, Uptime=00:03:16
       Output pkts=0, Input pkts=0
       METRIC DATA: Total rcvd=2, Avg arrival rate (ms)=10220
          CURRENT: MDR=2000000000000 bps, CDR=1000000000 bps
                   Lat=1 ms, Res=100, RLQ=90, load=1
          MDR      Max=2000000000000 bps, Min=2000000000000 bps, Avg=2000000000000 bps
          CDR      Max=1000000000 bps, Min=1000000000 bps, Avg=1000000000 bps
          Latency  Max=1, Min=1, Avg=1 (ms)
          Resource Max=100%, Min=100%, Avg=100%
          RLQ      Max=90, Min=90, Avg=90
          Load     Max=1%, Min=0%, Avg=0%
       Transport PPPoE, Session ID=2
       INTERFACE STATS:
          VMI Interface=vmi1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          V-Access intf=Virtual-Access1.1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          Physical intf=GigabitEthernet0/0/0,
             Input qcount=0, drops=0, Output qcount=0, drops=0

PPPoE Flow Control Stats
 Local Credits: 1954   Peer Credits: 65535   Local Scaling Value 65534 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 198     PADG Timer index: 283
 PADG last rcvd Seq Num: 0
 PADG last nonzero Seq Num: 283
 PADG last nonzero rcvd amount: 0
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 196  rcvd: 0
 PADC xmit: 128057344  rcvd: 196
 In-band credit pkt xmit: 0 rcvd: 0
 Last credit packet snapshot
  PADG xmit: seq_num = 198, fcn = 65535, bcn = 0
  PADC rcvd: seq_num = 198, fcn = 1954, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 105
  PADC xmit: seq_num = 109, fcn = 1951, bcn = 0
  In-band credit pkt xmit: fcn = 0, bcn = 0
  In-band credit pkt rcvd: fcn = 0, bcn = 0
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 1
Router-1#

Router#

Router-2#show vmi neighbor detail
              1 vmi1 Neighbors


vmi1   IPV6 Address=FE80::7E31:EFF:FE85:1E78
       IPV6 Global Addr=::
       IPV4 Address=71.0.0.1, Uptime=00:01:50
       Output pkts=0, Input pkts=0
       No Session Metrics have been received for this neighbor.
```

```
       Transport PPPoE, Session ID=1
       INTERFACE STATS:
          VMI Interface=vmi1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          V-Access intf=Virtual-Access1.1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          Physical intf=GigabitEthernet0/0/0,
             Input qcount=0, drops=0, Output qcount=0, drops=0

PPPoE Flow Control Stats
 Local Credits: 1953   Peer Credits: 65533  Local Scaling Value 65534 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 112     PADG Timer index: 168
 PADG last rcvd Seq Num: 0
 PADG last nonzero Seq Num: 168
 PADG last nonzero rcvd amount: 0
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 110  rcvd: 0
 PADC xmit: 127991808  rcvd: 110
 In-band credit pkt xmit: 0 rcvd: 0
 Last credit packet snapshot
  PADG xmit: seq_num = 112, fcn = 65535, bcn = 0
  PADC rcvd: seq_num = 112, fcn = 1953, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 70
  PADC xmit: seq_num = 71, fcn = 1952, bcn = 0
  In-band credit pkt xmit: fcn = 0, bcn = 0
  In-band credit pkt rcvd: fcn = 0, bcn = 0
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 0

Router-2#
```

Verify OSPF neighbor

This section shows examples of command output to verify your setup.

```
Router-1#sh ospfv3 neighbor | i FULL
102.102.102.102   0   FULL/  -        00:00:33    24    Virtual-Access1.1
102.102.102.102   0   FULL/  -        00:00:34    24          Virtual-Access1.1
Router-1

Router-2# sh ospfv3 neighbor | i FULL
sh ospfv3 neighbor | i FULL
101.101.101.101   0   FULL/  -        00:00:33    23          Virtual-Access1.1
101.101.101.101   0   FULL/  -        00:00:33    23          Virtual-Access1.1
Router-2#
```

Sample Running Configuration

This document displays a sample running configuration for a router, including interface, routing, and security settings.

```
Router-1#
show run
Building configuration...
Current configuration : 7911 bytes
```

```
!
! Last configuration change at 13:04:28 UTC Wed Jun 8 2022
!
version 17.9
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
platform hardware throughput level 250M
platform punt-keepalive disable-kernel-core
!
hostname Router-1
!
boot-start-marker
boot system flash bootflash:c6300-universalk9.17.09.01.SSA.bin
boot-end-marker
!
!
no logging console
no aaa new-model
!
!
login on-success log
```

ipv6 unicast-routing
ipv6 multicast-routing

```
!
!
```

subscriber templating
subscriber authorization enable

```
!
!
multilink bundle-name authenticated
!
!
crypto pki trustpoint TP-self-signed-4073554590
enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-4073554590
revocation-check none
rsakeypair TP-self-signed-4073554590
!
crypto pki trustpoint SLA-TrustPoint
enrollment pkcs12
revocation-check crl
!
!
crypto pki certificate chain TP-self-signed-4073554590
certificate self-signed 01
30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
69666963 6174652D 34303733 35353435 3930301E 170D3232 30363038 31313137
31345A17 0D333230 36303731 31313731 345A3031 312F302D 06035504 03132649
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D34 30373335
35343539 30308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
0A028201 0100C54E 860F73AC F0A9EBC7 2C6D1204 49099324 85989550 32CA9B91
3B3A2492 AEA1D550 0CD787DE 09F6B64F 3F01C578 3EFB3995 E448904A 957EAFD9
B82EF201 3A28BE26 6B1615DA 35B35BEA D4B7B20F 2D2A3EC2 C1F52281 349E88E5
F2BCBA37 CA72D461 97D1E817 7493ED38 7C7C1035 F7231D4E F59FADF9 EE0EE5EE
1FD73691 E93EBE8C 262DC8B1 0FA25BF0 C2F65BF9 C57A406A 9F9CF3D3 3E6888C3
D6B533AB 0DA71037 6C94A385 CECA4DD9 A037C344 5B761E6D F3B8D47B 4093BED8
```

```
E497D649 63436773 7BE5A718 331C7F08 31071542 03AE588C 08605CF7 CC7C7D6F
967759FC E2A943FF 8AD70094 825AAD4B DC66FEC8 5B7F2CDA 24F148E1 51AF106E
FE212C21 651B0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 14F49EA0 C5E78BB1 F0EA55AD 4C580FB7 BF1AE35C
50301D06 03551D0E 04160414 F49EA0C5 E78BB1F0 EA55AD4C 580FB7BF 1AE35C50
300D0609 2A864886 F70D0101 05050003 82010100 2083EBC1 E760806B 12F0ACEF
8FDE2E11 E5B88B62 B85B5835 AB1D2471 EA6FCC2D B28F5EE2 1969C233 DBB8C435
D7BAD49C 7781E485 97D5B5D0 DD05A0EE 5352535F 1657BE78 64E6BBA8 B627618E
49041DFB 4FE4D16C FA6857EF E6EEDFBC 2E25AF9E 852EDF71 3B65E55F 62AB1B1E
8B842F51 DAD55DB5 8A5BF87B 91F540D2 02E8576E 5D4550EA E7FCB6D7 6AD0E92A
EEAF7544 01C4095A BB02DB3A D45D73EB 971974FF B5DED058 F2F3A0E1 23BD3441
899CDFC2 A3B36E7D E72D4BDB 480B8347 C26D6AB9 E7E5A140 B20B7B1D 7AC24A2C
A69124B3 49BF18AF C99EF2A5 C4F484CE 9E2A70C1 D1EA4250 6E0D858E BBCFC6C1
4FF6E0BC EB190067 E86EC80D 5D149D6F 462CA857
quit
crypto pki certificate chain SLA-TrustPoint
certificate ca 01
30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
D697DF7F 28
quit
!
!
no license feature hseck9
license udi pid ESR-6300-CON-K9 sn FOC234304H3
```

license boot level network-advantage

```
memory free low-watermark processor 45135
!
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
!
redundancy
mode none
!
!
vlan internal allocation policy ascending
!
```

```
policy-map type service pppoe_rar
pppoe service manet_radio
!
!
   bba-group pppoe rar_group_1
           virtual-template 1
           service profile pppoe_rar

!
!
   interface Loopback0
           ip address 61.0.0.1 255.255.255.255
           ipv6 address 61::61/128
           ipv6 enable
           ospfv3 1 ipv6 area 0
           ospfv3 1 ipv4 area 0

!


           interface GigabitEthernet0/0/0
           media-type rj45
           negotiation auto
           ipv6 enable
           pppoe enable group rar_group_

1
!
interface GigabitEthernet0/0/1
no ip address
shutdown
negotiation auto
!
   interface GigabitEthernet0/1/0
           switchport access vlan 30

!
interface GigabitEthernet0/1/1
!
interface GigabitEthernet0/1/2
!
interface GigabitEthernet0/1/3
!
   interface Virtual-Template1
           mtu 1484
           ip unnumbered vmi1
           no ip redirects
           ip tcp adjust-mss 1444
           load-interval 30
           no peer default ip address
           ipv6 enable
           ipv6 mtu 1484
           ospfv3 1 network manet
```

```
            ospfv3 1 hello-interval 10
            ospfv3 1 ipv4 area 0
            ospfv3 1 ipv6 area 0

!
interface Vlan1
no ip address
!

    interface Vlan30
            ip address 192.168.10.1 255.255.255.0
            ipv6 address A000::1/64
            ipv6 enable
            ospfv3 1 ipv4 area 0
            ospfv3 1 ipv6 area 0

!
interface Async0/2/0
no ip address
encapsulation scada
!

    interface vmi1
            ip address 71.0.0.1 255.255.255.0
            ipv6 address FE80::7E31:EFF:FE85:1E78 link-local
            ipv6 address 71::71/64
            ipv6 enable
            physical-interface GigabitEthernet0/0/0
            mode bypass

!

    router ospfv3 1
            router-id 101.101.101.101

!

    address-family ipv4 unicast
            redistribute connected metric 1 metric-type 1
            log-adjacency-changes
            exit-address-family

!

    address-family ipv6 unicast
            redistribute connected metric-type 1
            log-adjacency-changes
            exit-address-family

!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
!

    ipv6 pim bsr candidate bsr 61::61
```

```
                                ipv6 pim bsr candidate rp 61::61

                        !
                        !
                        control-plane
                        !
                        !
                        mgcp behavior rsip-range tgcp-only
                        mgcp behavior comedia-role none
                        mgcp behavior comedia-check-media-src disable
                        mgcp behavior comedia-sdp-force disable
                        !
                        mgcp profile default
                        !
                        !
                        line con 0
                        exec-timeout 0 0
                        privilege level 15
                        stopbits 1
                        line 0/0/0
                        line 0/2/0
                        line vty 0 4
                        login
                        transport input ssh
                        line vty 5 14
                        login
                        transport input ssh
                        !
                        call-home
                        ! If contact email address in call-home is configured as
sch-smart-licensing@cisco.com
                        ! the email address configured in Cisco Smart License Portal will be used
as contact email address to send SCH notifications.
                        contact-email-addr sch-smart-licensing@cisco.com
                        profile "CiscoTAC-1"
                        active
                        destination transport-method http
                        !
                        end
```

Debug commands

This section shows debug commands for PPPoE, VMI, Virtual Template, and Subscriber Service.

PPPoE

Command or Action	Purpose
Router# **debug pppoe data**	PPPoE data packets
Router# **debug pppoe errors**	PPPoE protocol errors
Router# **debug pppoe events**	PPPoE protocol events
Router# **debug pppoe packets**	PPPoE control packets

VMI

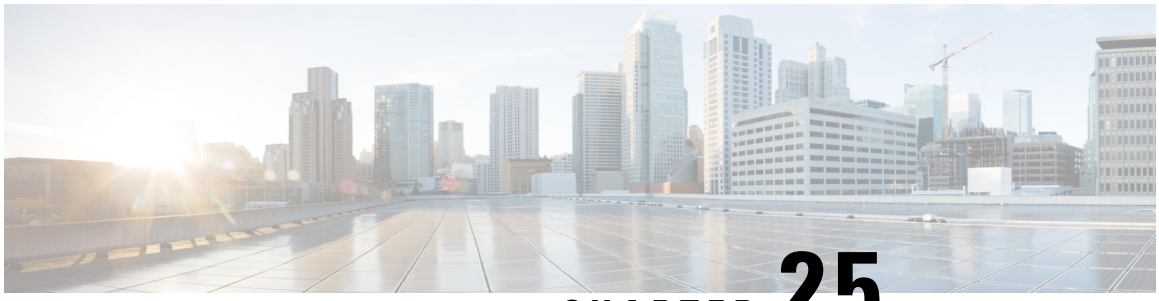
Command or Action	Purpose
Router# **debug VMI bma**	Display VMI bma debug
Router# **debug VMI error**	Display internal VMI anomalies detected
Router# **debug VMI multicast**	Display VMI multicast packets
Router# **debug VMI neighbor**	Display VMI neighbor transaction debugging
Router# **debug VMI packet**	Display VMI packet trace
Router# **debug VMI pppoe**	Display VMI PPPoE packet/activity debug
Router# **debug VMI registries**	Display VMI registry calls

Virtual template

Command or Action	Purpose
Router# **debug vtemplate cloning**	Virtual Template cloning information
Router# **debug vtemplate error**	Virtual Template errors
Router# **debug vtemplate event**	Virtual Template events
Router# **debug vtemplate subinterface**	Virtual Template subinterface command

Subscriber service

Command or Action	Purpose
Router# **debug sss errors**	Subscriber Service Switch Manager errors
Router# **debug ssss event**	Subscriber Service Switch Manager events



CHAPTER 25

Implementation Options

This section contains the following topics:

- SFP Overview, on page 285
- Power over Ethernet, on page 286
- Working with External USB3.0, on page 289

SFP Overview

The ESR6300 supports two WAN ports with RJ45 and optical/copper SFP ports.

The 2 Combo-WAN ports are g0/0/0 and g0/0/1. Each port can either be in Copper RJ45 or SFP mode, which is software configurable.

The SFP Ports have the following characteristics:

- Optical interface and Copper RJ45 are supported
- Whichever link is connected first is detected as default by software
- Auto-negotiation is supported for 10Mbps, 100Mbps and 1000Mbps
- Both Optical and Cu SFPs are supported, GE and FE alike
- Copper FE SFPs are not supported on ESR6300
- Auto-negotiation for 10Mbps, 100Mbps, 1000Mbps in full-duplex and half-duplex mode is supported. For half duplex, Cisco supports only 10Mbps and 100Mbps.
- Auto-select is supported. Whichever interface is detected first is selected. Typically fiber SFP will come up faster than RJ45 Copper, and is faster than Copper SFP.
- RJ45 to SFP mode conversion involves an interface reset (shut/no shut, interface DOWN/UP). The syslog can be observed accordingly.

Note Prior to release 17.2.1, Jumbo MTU support is not available on the ESR6300. MTU>1500 packets will be fragmented. From release 17.2.1 going forward, MTU support is 64 - 9216 on the interface. If there are bursts of fragmented traffic expected beyond the device Non-drop rate capability, software implements a throttling mechanism and transmits MAC Pause frames to the other end to reduce rate of traffic. To resume normal operation, arp has to sent again to re-negotiate. The default ip arp timeout is 14,000 seconds. In these rare case scenarios, it is recommended to have much shorter arp timeout configurations.

Supported SFPs are found in the ESR6300 Hardware Technical Guide here:

https://www.cisco.com/c/en/us/td/docs/routers/embedded/6300/hardware/tech-guide/ESR6300-tech-guide.html#pgfId-106538

SFP Command Line

To configure the WAN port as an RJ45 or an SFP:

```
Router#config terminal
Router(config)#config terminal
Router(config)#interface g0/0/{0|1}
Router(config-if)#media type ?
auto-select  Use whichever connector is attached
  rj45         Use RJ45 connector
  sfp          Use SFP connector
```

To configure auto-failover:

```
Router(config-if)#media-type {rj45|sfp} ?
  auto-failover  Automatic Fail over
  <cr>           <cr>
```

To validate your changes, use the following commands.

If SFP is detected:

```
Router#show inventory
```

To see if your configuration has taken effect:

```
Router#show run int g0/0/{0|1}
```

To reload the gigabit ethernet module:

```
Router#hwmodule subslot 0/0 reload force
```

Power over Ethernet

Power over Ethernet (PoE) is typically used to power up Access points, IP Cameras and IP Phones connected to the device's Ethernet ports.

The ESR6300 supports Power over Ethernet (PoE and PoE+) on the 4 LAN ports. PoE is NOT supported on the WAN ports. The power allocation is as follows:

- Supply 30W (PoE+) per port simultaneously on 4 LAN ports

- Supply 15.4W (PoE) per port simultaneously on 4 LAN ports
- The system integrator must have a 120W power supply



Note **The Powered Device (PD) will be detected if it is IEEE-compliance or a Cisco standard device. Support for CDP and LLDP is available for power negotiation, and must be enabled on the ESR6300**.



Note CDP and/or LLDP must be enabled on the ESR6300, and the PD must support CDP and/or LLDP for the device to be able to negotiate power levels between 15 and 30 watts.

Device Detection and Power Allocation

The router will detect a Cisco Pre-standard or an IEEE-compliant PD when the PoE is enabled and the connected device is not being powered by an AC adapter.

After device detection, the router determines the power requirements based on power classification class. Depending on the available power in the power budget, the router determines if a port can be powered. The router initially allocates this power when it detects and powers the device. Power negotiation using CDP/LLDP protocols happens thereafter.

Maximum power budget for 4 LAN ports combined at any time is 30W x4 = 120W. On reload the PoE ports are powered down.(i.e they are powered down at rommon stage).

CLI for PoE

This section describes the CLI to use for configuring and displaying PoE.

Before you configure Power over Ethernet (PoE), note the following:

- **show inventory** and **show diag** commands will not display details of the vendor/system integrator's PoE controller.
- **show run** command will not reflect the current PoE configuration.
- On connecting a PD, power negotiation happens almost instantly. However, it takes 3-5 minutes to reflect accurate statistics using **show power inline**
- The default software mode is PoE and not PoE+ to prevent overdraw.
- There is limited support for LLDP-MED and LLDP-MDI.



Note Implementation of PoE is a partner option. The integrator is responsible for proper implementation into the finished product, therefore, it may or may not be available.

To configure auto or off:

power inline *auto* | *never*

Configuration example:

```
Router#config terminal
Router#interface g0/1/<1,2,3,4>
Router(config-if)#power inline {auto|never}
```

To enable CDP:

```
Router#config terminal
Router(config)#cdp run
Router(config#exit
```

To enable LLDP:

```
Router#config terminal
Router(config)#lldp run
Router(config#exit
```

To Verify your configuration:

```
Router#show power inline
Available:120.0(w)  Used:21.1(w)  Remaining:98.9(w)

Interface Admin  Oper       Power   Device              Class Max
                            (Watts)
--------- ------ ---------- ------- ------------------ ----- ----
Gi0/1/0   auto   on         14.7    IP Phone 8865       4     30.0
Gi0/1/1   auto   on         6.3     IP Phone 8811       2     30.0
Gi0/1/2   auto   off        0.0     n/a                 n/a   30.0
Gi0/1/3   auto   off        0.0     n/a                 n/a   30.0
Router#
```

To show power on a particular interface:

```
Router#show power inline {interface-id}
```

Displays PoE status for a router for the specified interface.

show power inline *interface-id* detail

To show power consumption:

```
Router#show power
Main PSU :
    Total Power Consumption from 3.3V Line : 0.36
    Total Power Consumption from 5V Line : 6.20
    Configured Mode : N/A
    Current runtime state same : N/A
    PowerSupplySource : External PS
POE Module :
    Configured Mode : N/A
    Current runtime state same : N/A
    Total power available : 120 Watts
Router#
```

The list of commands for debugging PoE follows:

Command	Description
Debug ilpower controller	Display PoE controller debug messages

Command	Description
Debug ilpower event	Display PoE event debug messages
Debug ilpower port	Display PoE port manager debug messages
Debug ilpower powerman	Display PoE power management debug messages
Debug ilpower cdp	Display PoE CDP debug messages
Debug ilpower registries	Display PoE registries debug messages
Debug ilpower scp	Display PoE scp debug messages

Working with External USB3.0

The ESR6300 provides access to a single USB 3.0 Type A device.

External USB3.0

The following details are important when working with an external USB device:

- The USB is for storage only and can be gracefully mounted/unmounted using IOS CLI.
- The USB is accessible in ROMMON and IOS applications.
- USB device must have single partition, and in ext2, Fat16, or Fat32 format only.
- The user can copy files between usbflash0: to/from flash:/bootflash:
- In both ROMMON and IOS, use **dir usbflash0:** to view USB:



Caution No hot-plug support in rommon mode. On insertion of USB, reboot (rommon1>reset) to view usb.



Caution Cisco USBs are strongly recommended and are the only ones supported. Many generic USBs may not work. Some branded USBs which comply with protocol standards such as Kingston USB3.0 may work.

USB CLI Commands:

To access the USB file system through ROMMON, use the following command:

```
ROMMON>dir usbflash0:
```

To access the USB file system through IOS, use the following command:

```
Router#dir usbflash0:
```

To plug in and unplug the USB device gracefully, disable it first:

```
Router conf t
Router(config)#platform usb disable
Router#show platform usb status
USB disabled
```

To gracefully activate a mounted USB in IOS:

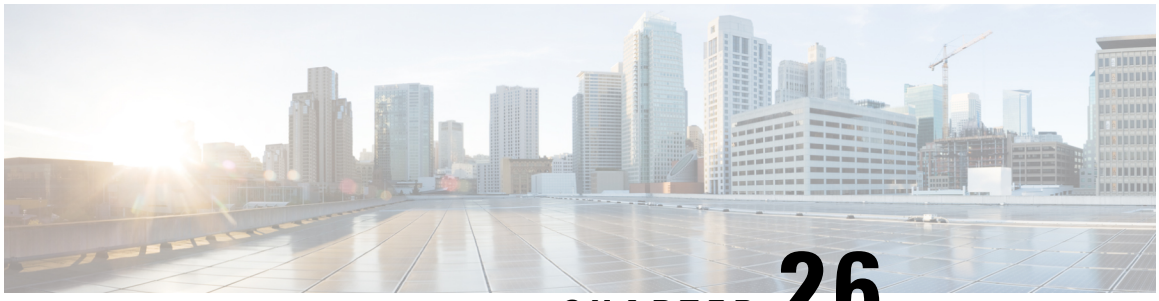
```
Router#no platform usb disable
Router#show platform usb status
USB enabled
```

The USB port could be considered a potential security risk. You may wish to disable it if it is not in use. To gracefully remove a USB when in IOS mode:

```
Router conf t
Router(config)#platform usb disable
Jun 4 05:44:52.339: %IOSD_INFRA-6-IFS_DEVICE_OIR: Device usbflash0 removed
Router#show platform usb status
USB disabled
```

To re-enable USB port:

```
Router(config)#no platform usb disable
*Jun 4 05:45:20.890: %IOSD_INFRA-6-IFS_DEVICE_OIR: Device usbflash0 added
Router#show platform usb status
USB enabled
```



CHAPTER 26

ROM Monitor Overview and Basic Procedures

This chapter provides an overview of ROM Monitor concepts and operations, and includes the following sections:

- ROM Monitor Overview, on page 291
- Access ROM Monitor Mode, on page 292
- Displaying the Configuration Register Setting, on page 294
- Environment Variable Settings, on page 295
- Exiting ROM Monitor Mode, on page 296

ROM Monitor Overview

The *ROM Monitor* is a bootstrap program that initializes the hardware and boots the Cisco IOS XE software when you power on or reload a router. When you connect a terminal to the router that is in ROM Monitor mode, the ROM Monitor (rommon 1>) prompt is displayed.

During normal operation, users do not use ROM Monitor mode. ROM Monitor mode is used only in special circumstances, such as reinstalling the entire software set, resetting the router password, or specifying a configuration file to use at startup.

The *ROM Monitor software* is known by many names. It is sometimes called *ROMMON* because of the CLI prompt in ROM Monitor mode. The ROM Monitor software is also called the *boot software* , *boot image* , or *boot helper* . Although it is distributed with routers that use the Cisco IOS XE software, ROM Monitor is a separate program from the Cisco IOS XE software. During normal startup, the ROM Monitor initializes the router, and then control passes to the Cisco IOS XE software. After the Cisco IOS XE software takes over, the ROM Monitor is no longer in use.

Rommon mode only allows for image boot via Flash: or USB (usbflash0:). There is no option to netboot.

Rommon mode has limitations on the USB variants it supports.

Note Cisco USBs are strongly recommended, and test validation support is claimed for only Cisco USBs. Many generic USBs may not work. Some branded USBs which comply with protocol standards such as Kingston USB3.0 may work.

Environmental Variables and the Configuration Register

Two primary connections exist between ROM Monitor and the Cisco IOS XE software: the ROM Monitor environment variables and the configuration register.

The ROM Monitor environment variables define the location of the Cisco IOS XE software and describe how to load it. After the ROM Monitor has initialized the router, it uses the environment variables to locate and load the Cisco IOS XE software.

The *configuration register* is a software setting that controls how a router starts up. One of the primary uses of the configuration register is to control whether the router starts in ROM Monitor mode or Administration EXEC mode. The configuration register is set in either ROM Monitor mode or Administration EXEC mode as needed. Typically, you set the configuration register using the Cisco IOS XE software prompt when you need to use ROM Monitor mode. When the maintenance in ROM Monitor mode is complete, you change the configuration register so the router reboots with the Cisco IOS XE software.

Accessing ROM Monitor Mode with a Terminal Connection

When the router is in ROM Monitor mode, you can access the ROM Monitor software only from a terminal connected directly to the console port of the card. Because the Cisco IOS XE software (EXEC mode) is not operating, nonmanagement interfaces are not accessible. Basically, all Cisco IOS XE software resources are unavailable. The hardware is available, but no configuration exists to make use of the hardware.

Network Management Access and ROM Monitor Mode

It is important to remember that ROM Monitor mode is a router mode, not a mode within the Cisco IOS XE software. It is best to remember that ROM Monitor software and the Cisco IOS XE software are two separate programs that run on the same router. At any given time, the router runs only one of these programs, .

One area that can be confusing when using ROM Monitor and the Cisco IOS XE software is the area that defines the IP configuration for the Management Ethernet interface. Most users are comfortable with configuring the Management Ethernet interface in the Cisco IOS XE software. When the router is in ROM Monitor mode, however, the router does not run the Cisco IOS XE software, so that Management Ethernet interface configuration is not available.

When you want to access other devices, such as a TFTP server, while in ROM Monitor mode on the router, you must configure the ROM Monitor variables with IP access information.

Access ROM Monitor Mode

The following sections describe how to enter the ROMMON mode.

Checking the Current ROMMON Version

To display the version of ROMmon running on a router, use the **show rom-monitor** command . To show all variables that are set in ROMmon, use show romvar.

```
Router#show rom-monitor r0
System Bootstrap, Version 2.1(REL), RELEASE SOFTWARE
Copyright (c) 1994-2019  by cisco Systems, Inc

Router# show romvarROMMON variables:
 PS1 = rommon ! >
```

```
 MCP_STARTUP_TRACEFLAGS = 00000000:00000000
 THRPUT = 50
 LICENSE_BOOT_LEVEL = network-essentials,network-stack:esg;
 RET_2_RTS =
 BSI = 0
 RET_2_RCALTS =
 RANDOM_NUM = 1027275747

Router# reload
```

Default manufacturing config-register value in rommon> is 0x102 [baud rate 9600]. If you would like to change the config-register value, it must be changed separately in both rommon> and IOS modes.

```
rommon 1 > set
PS1=rommon ! >
MCP_STARTUP_TRACEFLAGS = 00000000:00000000
 LICENSE_SUITE =
 RET_2_RTS =
 Diagnostic = 1
 THRPUT =
 USER_BOOT_PARAM =  DEBUG_CONF=/bootflash/debug.conf
 EULA_ACCEPTED = TRUE
 BOOT_WDOG = DISABLE
 LICENSE_BOOT_LEVEL =
 BOOT = bootflash:sparrow_crashkernel.bin,1;
 CRASHINFO = bootflash:crashinfo_RP_00_00_20180619-204307-UTC
 RET_2_RCALTS =
 BSI = 0
 RANDOM_NUM = 1662155698
```

Commonly Used ROM Monitor Commands

The following table summarizes the commands commonly used in ROM Monitor. For specific instructions on using these commands, refer to the relevant procedure in this document.

Table 22: Commonly Used ROM Monitor Commands

ROMMON Command	Description
boot image	Manually boots a Cisco IOS XE software image.
boot image –o config-file-path	Manually boots the Cisco IOS XE software with a temporary alternative administration configuration file.
confreg	Changes the config-register setting.
dev	Displays the available local storage devices.
dir	Displays the files on a storage device.
reset	Resets the node.
set	Displays the currently set ROM Monitor environmental settings.
sync	Saves the new ROM Monitor environmental settings.
unset	Removes an environmental variable setting.

ROMMON Command Examples

The following example shows what appears when you enter the **?** command on a router:

```
rommon 1 > ?
alias               set and display aliases command
boot                boot up an external process
confreg             configuration register utility
dev                 list the device table
dir                 list files in file system
help                monitor builtin command help
history             monitor command history
meminfo             main memory information
repeat              repeat a monitor command
reset               system reset
set                 display the monitor variables
showmon             display currently selected ROM monitor
sync                write monitor environment to NVRAM
token               display board's unique token identifier
unalias             unset an alias
unset               unset a monitor variable
```

Changing the ROM Monitor Prompt

You can change the prompt in ROM Monitor mode by using the **PS1=** command as shown in the following example:

```
rommon 8 > PS1="ESR6300 rommon ! > "
ESR6300 rommon 9 >
```

Changing the prompt is useful if you are working with multiple routers in ROM Monitor at the same time. This example specifies that the prompt should be “ESR6300 rommon ”, followed by the line number, and then followed by “ > “ by the line number.

Displaying the Configuration Register Setting

To display the current configuration register setting, enter the **confreg** command without parameters as follows:

```
rommon > confreg
 Configuration Summary
   (Virtual Configuration Register: )
enabled are:
[ 0 ] break/abort has effect
[ 1 ] console baud: 9600
 boot:...... the ROM Monitor
do you wish to change the configuration? y/n  [n]:
```

The configuration register setting is labeled *Virtual Configuration Register* . Enter the **no** command to avoid changing the configuration register setting.

Environment Variable Settings

The ROM Monitor environment variables define the attributes of the ROM Monitor. Environmental variables are entered like commands and are always followed by the equal sign (=). Environment variable settings are entered in capital letters, followed by a definition. For example:

```
IP_ADDRESS=10.0.0.2
```

Under normal operating conditions, you do not need to modify these variables. They are cleared or set only when you need to make changes to the way ROM Monitor operates.

This section includes the following topics:

Frequently Used Environmental Variables

The following table shows the main ROM Monitor environmental variables. For instructions on how to use these variables, see the relevant instructions in this document. The ESR6300 boot loader does not support netboot, so any setting like environment variables IP_ADDRESS, IP_SUBNET_MASK, DEFAULT_GATEWAY, TFTP_SERVER, TFTP_FILE are not used.

Table 23: Frequently Used ROM Monitor Environmental Variables

Environmental variable	Description
BOOT=path/file	Identifies the boot software for a node. This variable is usually set automatically when the router boots.

Displaying Environment Variable Settings

To display the current environment variable settings, enter the **set** command :

```
rommon 1 > showmon
System Bootstrap, Version 1.3(REL), RELEASE SOFTWARE
Copyright (c) 1994-2018 by cisco Systems, Inc.

ESR 6300-K9 platform with 4188160 Kbytes of main memory

MCU Version - Bootloader: 4, App: 4
MCU is in application mode.
```

Entering Environment Variable Settings

Environment variable settings are entered in capital letters, followed by a definition. The following example shows the environmental variables that can be configured in ROMmon mode.:

```
rommon 1> confreg 0x0
rommon 1> BOOT_WDOG = DISABLE
rommon 1> BOOT = ESR6300-K9_image_name
```

Saving Environment Variable Settings

To save the current environment variable settings, enter the **sync** command:

```
rommon > sync
```



Note Environmental values that are not saved with the **sync** command are discarded whenever the system is reset or booted.

Exiting ROM Monitor Mode

To exit ROM Monitor mode, you must change the configuration register and reset the router.

Procedure

	Command or Action	Purpose
Step 1	**confreg** **Example:** `rommon 1> confreg`	Initiates the configuration register configuration prompts.
Step 2	Respond to each prompt as instructed.	See the example that follows this procedure for more information.
Step 3	**reset** **Example:** `rommon 2> reset`	Resets and initializes the router.

Configuration Example

```
rommon 3 > confreg
          Configuration Summary
   (Virtual Configuration Register: 0x0)
enabled are:
 [ 0 ] break/abort has effect
 [ 1 ] console baud: 9600
 boot: ...... the ROM Monitor
do you wish to change the configuration? y/n  [n]:  y
 enable  "diagnostic mode"? y/n  [n]:
 enable  "use net in IP bcast address"? y/n  [n]:
 enable  "load rom after netboot fails"? y/n  [n]:
 enable  "use all zero broadcast"? y/n  [n]:
 disable "break/abort has effect"? y/n  [n]:
 enable  "ignore system config info"? y/n  [n]:
 change console baud rate? y/n  [n]:
change the boot characteristics? y/n  [n]:
```

```
          Configuration Summary
   (Virtual Configuration Register: 0x0)
enabled are:
 [ 0 ] break/abort has effect
 [ 1 ] console baud: 9600
 boot: ...... the ROM Monitor
do you wish to change the configuration? y/n  [n]:
```

Upgrading the ROMMON for a Router

ROMMON upgrade on the ESR6300 router is automatically done when the image is booted. The latest version of the ROMMON is bundled with the IOS XE image. An algorithm detects if the current running version is older than the bundled version, if so, it is automatically upgraded. If the current running version is equal to the bundled version no upgrade is executed. For every successful upgrade, the router is automatically rebooted in order for the new version to get loaded and executed.:

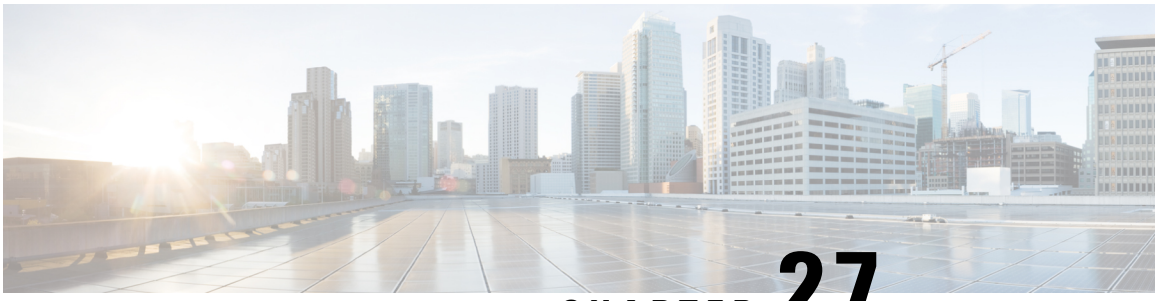
Procedure

Step 1 (Optional) Run the **show rom-monitor** *slot* command on the router to see the current release numbers of ROMMON on the hardware. See the Checking the Current ROMMON Version, on page 292 for information about interpreting the output of the command that you run.

Step 2 If autoboot has not been enabled by using the **config-register 0x2102** command, run the **boot** *filesystem:/file-location* command at the ROMMON prompt to boot the Cisco IOS XE image, where *filesystem:/file-location* is the path to the consolidated package file. The ROMMON upgrade is not permanent for any piece of hardware until the Cisco IOS XE image is booted.

Step 3 Run the **enable** command at the user prompt to enter the privileged EXEC mode after the boot is complete.

Step 4 Run the **show rom-monitor** *slot* command to verify whether the ROMMON has been upgraded.



CHAPTER 27

Device Zeroization

This section contains the following topics:

- Push Button, on page 299
- Device Zeroization, on page 300

Push Button

There is no actual button on the ESR6300, and the system integrator must configure their platform with a Push Button. Reset on an ESR6300 does not cause the device to reboot, but initiates the configured level of zeroization.

- When the system is running in IOS mode, pressing this push button for 4+ seconds will cause files erase in flash, and will reset to factory-default mode on boot up.

  The button must be pressed while the system is turned on at the same time.

  The push button must continue to be held for more than 4 seconds after the power is turned on.

  Config-reg setting is in NVRAM, and not changed by the button push.

Pressing the push button when in rommon mode has no effect.

Pressing the push button when in IOS mode causes a syslog message to appear and triggers a reload.

Pressing the push button for more than 4+ seconds after power up displays the following message when reset has been triggered:

```
System Bootstrap, Version 1.4(DEV) [vandvisw-vandvisw 113], DEVELOPMENT SOFTWARE
Copyright (c) 1994-2019 by cisco Systems, Inc.
Compiled at Mon Jun 3 10:56:19 2019 by vandvisw
ESR-6300-CON-K9 platform with 4194304 Kbytes of main memory
MCU Version - Bootloader: 8, App: 10
MCU is in application mode.
Reset button push detected
```

How the router boots up depends on the following conditions:

- If there is a golden image in flash, the router will bootup with this image by default.
- If a golden image is not found, the router will be in rommon> mode as expected. Or, based on config-reg entry, may boot up first image in the flash:

Microcontroller Unit (MCU)

The MCU is part of the ESR6300 hardware. It performs the following functions:

- Monitors the Push button status at power up
- Monitors the system hardware watchdog output
- Maintains Reset Reason register
- Controls the SYS LED

The MCU versions are displayed using **show version**. Details on MCU version and upgrade status are also stored in Flash: as boothelper.log. The MCU is automatically upgraded by the software.

```
Router#show ver | i MCU
MCU bootloader version: 8
MCU application version: 10

Router#cat flash:boothelper.log
Logging at Fri Nov 15 05:00:54 Universal 2019
boot loader upgrade enabled
Bootloader is up-to-date
Current MCU App version is 10
MCU firmware is up-to-date
```

In the event the MCU Application is corrupt, or does not match the Release Notes version, this has to be repaired. Steps to recover from this state: Reload router, hit Ctrl+C to break into rommon mode.

```
Rommon>set MCU_UPGRADE=IGNORE - Ignore MCU firmware upgrade errors.
Rommon>sync
Rommon>reset
Rommon>boot bootflash:<image>
```

Once the MCU successfully upgrades, you can disable/unset this IGNORE option in rommon. Details on other MCU setting rommon options follow: (there are no available IOS configuration options or linux shell mode troubleshooting measures)

```
set MCU_UPGRADE=SKIP – Prevents MCU firmware upgrade from taking place.
set MCU_UPGRADE=FORCE – Forces MCU firmware upgrade to take place.
unset MCU_UPGRADE – Normal operation. Allows automatic upgrade
```

Device Zeroization

Zeroization consists of erasing any and all potentially sensitive information in the device.

This includes erasure of Main memory, cache memories, and other memories containing packet data, NVRAM, and Flash memory. The process of zeroization is launched upon the initiation of a user command and a subsequent trigger.

By default, the router will have the zeroization feature disabled. SPI: Flash, I2C, and ACT2 are not impacted by this feature.

When zeroization is functionally active, SYS LED indicates blinking yellow until the router reloads.

Zeroization can be triggered by the push button, or software-triggered by a privilege 15 user with console access. There is no remote access for security reasons. On triggering zeroization, the eMMC, NVRAM will be erased completely.

The zeroization process starts as soon as the push button is pressed down or the command is triggered.. The CLI command, "service declassify", is used to set the desired action in response to push button press. To prevent accidental erasure of the system configuration/image, the default setting is set to "no service declassify".

Note For complete details on how the process of zeroization works, see the Cisco Embedded Service 6300 Series Router Hardware Technical Guide.

Zeroization Trigger

Zeroization can be triggered by either software or by the push button. In either case, there are a series of commands that need to be entered.

```
Router#config terminal
Router(config)#service declassify {erase-nvram | erase-all}
```

In the above example, erase-nvram will wipe only nvram, while erase-all will wipe bootflash: and nvram as well.

To confirm if the feature is enabled:

```
Router#show declassify
Declassify facility: Enabled=Yes  In Progress=No
                     Erase flash=Yes  Erase nvram=Yes
  Declassify Console and Aux Ports
  Shutdown Interfaces
  Reload system
```

To remove the feature, use the following command:

```
Router(config)#no service declassify
```

To Trigger Zeroization

To trigger the zeroization from the command line:

```
Router#declassify trigger
```

To trigger the zeroization from the push button, press and hold the button for 4+ seconds. When the system auto reloads, it will come up in ROMMON mode: "$$" with bootflash: wiped clean.

Important Notice about Zeroization

eMMC is a managed NAND. This means that our router system does not interact with the flash memory directly. The flash controller presents a block-style interface to our system, and it handles the flash management (analogous to the Flash Translation Layer). Our embedded router system cannot access the raw flash directly.

The JEDEC standard has commands that are supposed to remove data from the raw flash. In Cisco's implementation, the "Erase" and "Sanitize" commands are used. The eMMC standard JESD84-B51 defines "Sanitize" as follows:

The Sanitize operation is a feature … that is used to remove data from the device according to Secure Removal Type. The use of the Sanitize operation requires the device to physically remove data from the unmapped user address space.

After the sanitize operation is completed, no data **should exist**in the unmapped host address space.

Warning Zeroization does NOT erase removable media such as SD Card and USB Storage. This media must be removed from the system and erased or destroyed using procedures that are outside the scope of this document.

Zeroize does a very thorough wipe of all non-protected parts of the eMMC flash using the best technology designed by the flash manufacturer today and can do so using the push of a button without the need for a console, ssh, or management session of any kind. It is the integrator's and end user's responsibility to determine the suitability regardless of the CLI keyword used to enable the feature.

Note **While Cisco IOS and Cisco IOS-XE use the command line text of "declassify" in the command line interface (CLI) to enable the zeroize feature, in no way does this represent any specific endorsement or acknowledgment of a Government approved flash erasure methodology.**

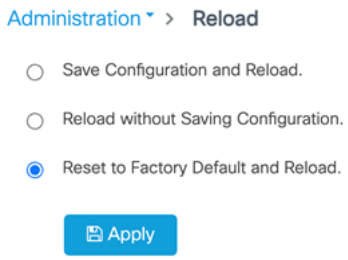
Declassification procedures are unique to each Government organization. Cisco solely provides the technical detail of the erasure operation here, not the policy distinction or any specific recommendation per classification.

Please refer to your respective Government Agency policies, procedures, and recommendations for the handling of sensitive data to see if this procedure meets with those requirements.

WARNING!

The CLI **service declassify erase-all** is literally a **software self-destruct mechanism** intended for defense and intelligence environments that attempts to wipe clean, all of the writable non-volatile storage on the device to clear the device configuration, other stored configurations and all security credentials including any additional license keys.

Please do not use this feature in lieu of doing a **write erase** from the CLI or from the Administration page, Reload option of the WebUI. Invoke the reload with the **Reset to Factory Default and Reload** option and click **Apply**. See the following figure.



If **service declassify erase-all** is invoked, after restoring the IOS-XE image and device configuration, you must re-license the device using the standard Cisco Smart Licensing procedures which ultimately require a Cisco Smart Account and access to the internet or a satellite license server.

Command Line Interface

There are two levels of zeroization actions, **erase-nvram** and **erase-all**. The following CLI shows the options:

```
device(config)#service declassify ?
erase-nvram    Enable erasure of device configuration as zeroization action. Default is no
 erasure.
erase-all      Enable erasure of both flash and nvram file systems as part of zeroization.
 Default is no erasure
```

The perma-locked bootable image(s) in the flash file system will still be available and can be used for booting the device.

The “erase-all” level of zeroization process erases the entire flash file system. This also wipes out all files and perma-locked bootable image(s). All interfaces are shut down before this process. Here, erasure of individual files in the flash file system is not possible and the only option is to erase the entire flash file system. This also erases packet data, ASIC data and processors related caches along with scrubbing Main memory.

With any level of zeroization, the device always fall back to the ROMMON prompt on the console after the erasure of configuration files or flash file system.

Zeroization Support in bootloader

The zeroization process may take several minutes, depending on several system parameters such as the size of DDR memory, EMMC disk size, etc.

It is possible that the zeroization may get interrupted by a power cycle before it completes. Since the primary OS image on EMMC itself gets purged during zeroization, it becomes impossible to continue zeroization after a power cycle. To solve this, zeroization support has been in the bootloader and will run it to completion even if it gets interrupted by power cycles.

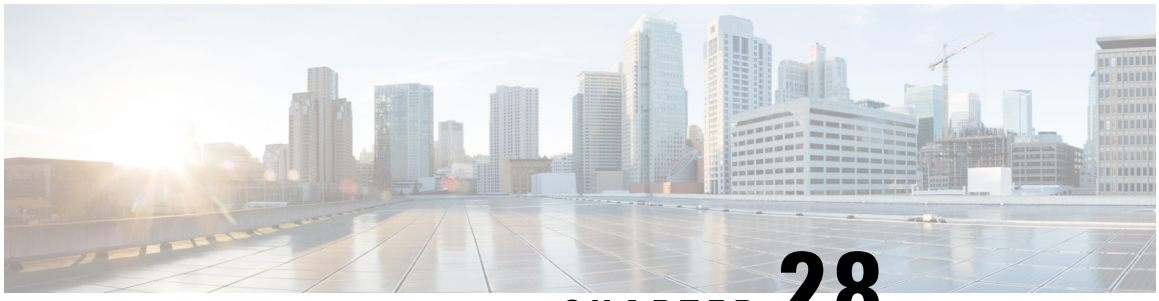
Troubleshooting Zeroization

The following table shows troubleshooting tips for zeroization:

Table 24:

Symptoms	Troubleshooting
Upon zeroization, some files are still left behind in the bootflash:	In that case, collect logs to pass on to Cisco support/engineering teams for further debugging. Upon zeroization, the ideal state of the router should be in rommon mode displayed as $. When executing $dir bootflash:, the partition should not even be mounted.
How do you know if zeroization was triggered if the router is in $$ prompt?	Symptoms are bootflash: unmounted, router should be stuck in rommon and environment variables [rommon>set] should be wiped clean. There is no syslogging or reset reason displayed.

Symptoms	Troubleshooting
Zeroization does not take effect on trigger.	Check MCU version: **show ver | I MCU**. MCU application version should reflect the latest as per the Release Notes. If it is anything else, likely Application is corrupt and we need to re-trigger MCU upgrade. Capture the output from **show platform declassify**. Ensure you have actually enabled zeroization, before executing **declassify trigger**. Ensure you have privilege 15 to trigger zeroization. Ensure you are on console access. SSH/telnet remote session zeroization trigger is considered a security issue and is therefore disabled.



CHAPTER 28

Call Manager Express

This section contains the following topics:

- Call Manager Express Overview, on page 305
- SCCP Configuration Example, on page 305
- SCCP Show Command Example, on page 306
- SIP Configuration Example, on page 307
- Debug Commands, on page 308
- SIP Show Command Example, on page 308

Call Manager Express Overview

Cisco Unified Communications Manager Express (formerly known as Cisco Unified Call Manager Express) is a call-processing application in Cisco IOS software that enables Cisco routers to deliver key-system or hybrid PBX functionality for enterprise branch offices or small businesses.

The Network-Advantage license is required for the CME feature on the ESR6300:

IP Phones are powered by ESR6300 LAN ports PoE/PoE+ and can negotiate power if they support the CDP/LLDP protocols.

There are several sources of information available for Call Manager Express:

- Product Marketing Information
- Cisco Unified Communications Manager Express System Administrator Guide
- All of the Support Information
- Basic Configuration Guide for CUCME/CME
- Cisco Unified Communications Manager Express FAQ

SCCP Configuration Example

Generate configuration files for SCCP Phones:

```
telephony-service
max-ephones 24
```

```
max-dn 200
ip source-address 192.168.8.15 port 2000
load 7945 term45.default.loads
!
!
ephone-dn  8
number 1008
name sccp-08
!
!
ephone-dn  9
number 1009
name sccp-09
!
ephone  8
device-security-mode none
mac-address 00A3.D1F3.6FAD
type 7945
button  1:8
!
!
ephone  9
device-security-mode none
mac-address 00A3.D1F3.6FC0
type 7945
button  1:9
```

SCCP Configuration tftp-server:

```
tftp-server bootflash:term45.default.loads
tftp-server bootflash:SCCP45.9-4-2SR3-1S.loads
tftp-server bootflash:cvm45sccp.9-4-2ES26.sbn
tftp-server bootflash:cnu45.9-4-2ES26.sbn
tftp-server bootflash:apps45.9-4-2ES26.sbn
tftp-server bootflash:jar45sccp.9-4-2ES26.sbn
tftp-server bootflash:dsp45.9-4-2ES26.sbn
!
```

SCCP Show Command Example

The following are two examples of SCCP:

```
Router#sh telephony-service ephone
Number of Configured ephones 2 (Registered 0)
ephone 8
Device Security Mode: Non-Secure
mac-address 00A3.D1F3.6FAD
type 7945
button  1:8
keepalive 30 auxiliary 30
actual keepalive 30 auxiliary 30
multicast-moh
max-calls-per-button 8
busy-trigger-per-button 0
Always send media packets to this router: No
Preferred codec: g711ulaw
conference drop-mode never
conference add-mode all
conference admin: No
privacy: Yes
privacy button: No
```

```
user-locale US
network-locale US
lpcor type:
lpcor (incoming):             (outgoing):
!
ephone 9
Device Security Mode: Non-Secure
mac-address 00A3.D1F3.6FC0
type 7945
button  1:9
keepalive 30 auxiliary 30
actual keepalive 30 auxiliary 30
multicast-moh
max-calls-per-button 8
busy-trigger-per-button 0
Always send media packets to this router: No
Preferred codec: g711ulaw
conference drop-mode never
conference add-mode all
conference admin: No
privacy: Yes
privacy button: No
user-locale US
network-locale US
lpcor type:
lpcor (incoming):             (outgoing):
!
Router#sh telephony-service ephone-dn

ephone-dn 8
number 1008
name sccp-08
preference 0 secondary 9
huntstop
call-waiting beep
no cti notify
no cti watch
ephone-dn 9
number 1009
name sccp-09
preference 0 secondary 9
huntstop
call-waiting beep
no cti notify
no cti watch
```

SIP Configuration Example

Generate Configuration Profiles for SIP Phones:

```
voice register global
mode cme
source-address 192.168.8.16 port 5060
max-dn 20
max-pool 15
load 8845 sip8845_65.12-5-1SR2-2.loads
tftp-path flash:
file text
create profile sync 0095724401230545
ntp-server 192.168.8.15 mode unicast
auto-register
!
```

```
voice register dn  2
number 202
call-forward b2bua busy 100
 call-forward b2bua noan 100 timeout 20
allow watch
name SIP-PHONE2
label 4085252002
voice register pool  2
busy-trigger-per-button 2
id mac 0045.1D86.F992
type 8845
number 1 dn 2
dtmf-relay sip-notify
codec g711ulaw
```

SIP Configuration TFTP-Server:

```
tftp-server flash:vc48845_65.12-5-1SR2-2.sbn
tftp-server flash:fbi8845_65.BEV-01-006.sbn
tftp-server flash:kern8845_65.12-5-1SR2-2.sbn
tftp-server flash:rootfs8845_65.12-5-1SR2-2.sbn
tftp-server flash:sb28845_65.BEV-01-015.sbn
tftp-server flash:sip8845_65.12-5-1SR2-2.loads
!
```

Debug Commands

Two commands that are helpful in debugging are:

```
router#debug ccsip message
router#debug voice register event
router#
000313: *Jun 18 02:54:31.888: voice_reg_process_bulk_registration: preparing unsolicited
sharedln registration for 6010@28.18.88.70
000314: *Jun 18 02:54:31.888: voice_reg_process_bulk_registration: preparing unsolicited
sharedln registration for 6779@28.18.88.70
000315: *Jun 18 02:54:31.888: voice_reg_process_bulk_registration: preparing unot presence
 registration for watcher [sip:6010@28.18.88.70] presentity [sip:6777@28.18.88.1]
000316: *Jun 18 02:54:31.888: voice_reg_process_bulk_registration: preparing unot presence
 registration for watcher [sip:6010@28.18.88.70] presentity [sip:6506777@28.18.88.1]
```

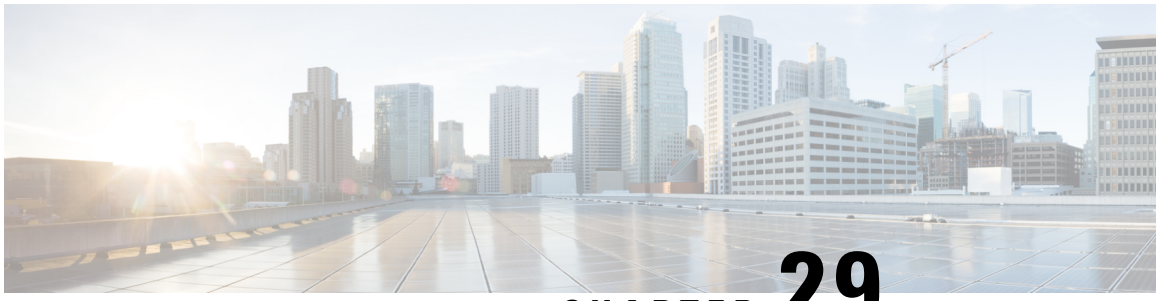
SIP Show Command Example

The following are two examples of SIP:

```
Router#sh voice register global
CONFIG [Version=12.6]
=======================
  Version 12.6
  Mode is cme
  Auto-registration is enabled
  Max-pool is 15
  Max-dn is 20
  Outbound-proxy is enabled and will use global configured value
  Security Policy: DEVICE-DEFAULT
  Forced Authorization Code Refer is enabled
  Source-address is 192.168.8.16 port 5060
  Load 8845 is sip8845_65.12-5-1SR2-2.loads
```

```
  NTP Server Address is 192.168.8.15
  NTP Server Mode is: unicast
  Time-format is 12
  Total SIP phones registered: 1
  Total Registration Statistics
    Registration requests  : 8
    Registration success   : 2
    Registration failed    : 0
    unRegister requests    : 1
    unRegister success     : 1
    unRegister failed      : 0
    Auto-Register requests : 0
    Attempts to register
           after last unregister : 0
    Last register request time   : 11:29:38.113 PDT Mon Jul 8 2019
    Last unregister request time : 07:16:06.845 PDT Wed Jul 3 2019
    Register success time        : 11:29:38.113 PDT Mon Jul 8 2019
    Unregister success time      : 07:16:06.845 PDT Wed Jul 3 2019

Router#sh voice register dn 2
Dn Tag 2
Config:
  Number is 202
  Preference is 0
  Huntstop is disabled
  Name SIP-PHONE2
  Allow watch is enabled
  Auto answer is disabled
  Label is 4085252002
  call-forward b2bua busy 100
  call-forward b2bua noan 100 timeout 20
  Pool 2    has this DN configured for line 1
```



CHAPTER 29

Troubleshooting

This chapter contains the following topics:

- Getting Started with Troubleshooting, on page 311
- Understanding Diagnostic Mode, on page 311
- Before Contacting Cisco or Your Reseller, on page 312
- show interfaces Troubleshooting Command, on page 312
- Software Upgrade Methods, on page 312
- Recovering a Lost Password, on page 313

Getting Started with Troubleshooting

Before troubleshooting a software problem, you must connect a PC to the router via the console port. With a connected PC, you can view status messages from the router and enter commands to troubleshoot a problem.

You can also remotely access the interface by using Telnet. The Telnet option assumes that the interface is up and running.

Understanding Diagnostic Mode

The router boots up or accesses diagnostic mode in the following scenarios:

- The IOS process or processes fail, in some scenarios. In other scenarios, the system resets when the IOS process or processes fail.
- A user-configured access policy was configured using the **transport-map** command that directs the user into the diagnostic mode.
- A send break signal (**Ctrl-C** or **Ctrl-Shift-6**) was entered while accessing the router, and the router was configured to enter diagnostic mode when a break signal was sent.

In the diagnostic mode, a subset of the commands that are available in user EXEC mode are made available to the users. Among other things, these commands can be used to:

- Inspect various states on the router, including the IOS state.
- Replace or roll back the configuration.
- Provide methods of restarting the IOS or other processes.

- Reboot hardware, such as the entire router, a module, or possibly other hardware components.
- Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

The diagnostic mode provides a more comprehensive user interface for troubleshooting than previous routers, which relied on limited access methods during failures, such as ROMMON, to diagnose and troubleshoot Cisco IOS problems. The diagnostic mode commands can work when the Cisco IOS process is not working properly. These commands are also available in privileged EXEC mode on the router when the router is working normally.

Before Contacting Cisco or Your Reseller

If you cannot locate the source of a problem, contact your local reseller for advice. Before you call, you should have the following information ready:

- Chassis type and serial number
- Maintenance agreement or warranty information
- Type of software and version number
- Date you received the hardware
- Brief description of the problem
- Brief description of the steps you have taken to isolate the problem

show interfaces Troubleshooting Command

Use the **show interfaces** command to display the status of all physical ports and logical interfaces on the router.

The ESR6300 supports the following interfaces:

GigabitEthernet combo ports0/0/0 and 0/0/1

Gigabit Ethernet LAN ports g0/1/0-4

Pluggable Cellular Module 0/3/0 and 0/3/1

Software Upgrade Methods

Several methods are available for upgrading software on the Cisco ESR6300 Routers, including:

- Copy the new software image to flash memory over LAN or WAN when the existing Cisco IOS software image is in use.
- Copy the new software image to flash memory over the LAN while the boot image (ROM monitor) is operating.
- Copy the new software image over the console port while in ROM monitor mode.

- From ROM monitor mode, boot the router from a software image that is loaded on a TFTP server. To use this method, the TFTP server must be on the same LAN as the router.

Recovering a Lost Password

To recover a lost enable or lost enable-secret password, refer to the following sections:

1. Change the Configuration Register
2. Reset the Router
3. Reset the Password and Save your Changes (for lost enable secret passwords only)
4. Reset the Configuration Register Value.

5. If you have performed a **write erase**, or used the reset button, you will need to add the license.

```
Router#config term
Router#license smart reservation
```

Note Recovering a lost password is only possible when you are connected to the router through the console port. These procedures cannot be performed through a Telnet session.

Tip See the "Hot Tips" section on Cisco.com for additional information on replacing enable secret passwords.

Change the Configuration Register

To change a configuration register, follow these steps:

Procedure

Step 1 Connect a PC to the CONSOLE port on the router.

Step 2 At the privileged EXEC prompt (r*outer_name* #), enter the **show version** command to display the existing configuration register value (shown in bold at the bottom of this output example):

Example:

```
Router# show version
Cisco IOS XE Software, Version 17.01.01
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2019 by Cisco Systems, Inc.
Compiled Tue 19-Nov-19 06:19 by mcpre

(output omitted for brevity)
Configuration register is 0x2102
Router#
```

Step 3 Record the setting of the configuration register.

Step 4 To enable the break setting (indicated by the value of bit 8 in the configuration register), enter the **config-register 0x01** command from privileged EXEC mode.

- Break enabled—Bit 8 is set to 0.
- Break disabled (default setting)—Bit 8 is set to 1.

Configuring the Configuration Register for Autoboot



Note Altering the configuration register is only for advanced troubleshooting and should only be done with guidance from Cisco support.

The configuration register can be used to change router behavior. This includes controlling how the router boots. Set the configuration register to 0x0 to boot into ROM, by using one of the following commands:

- In Cisco IOS configuration mode, use the **config-reg** 0x0 command.
- From the ROMMON prompt, use the **confreg** 0x0 command.



Note Setting the configuration register to 0x2102 will set the router to autoboot the Cisco IOS XE software.

Reset the Router

To reset the router, follow these steps:

Procedure

Step 1 If break is disabled, turn the router off (O), wait 5 seconds, and turn it on (|) again. Within 60 seconds, press the **Break** key. The terminal displays the ROM monitor prompt.

Note
Some terminal keyboards have a key labeled *Break* . If your keyboard does not have a Break key, see the documentation that came with the terminal for instructions on how to send a break.

Step 2 Press break. The terminal displays the following prompt:

Example:

```
rommon 2>
```

Step 3 Enter **confreg 0x142** to reset the configuration register:

Example:

```
rommon 2> confreg 0x142
```

Step 4 Initialize the router by entering the **reset** command:

Example:

```
rommon 2> reset
```

The router cycles its power, and the configuration register is set to 0x142. The router uses the boot ROM system image, indicated by the system configuration dialog:

Example:

```
--- System Configuration Dialog ---
```

Step 5 Enter **no** in response to the prompts until the following message is displayed:

Example:

```
Press RETURN to get started!
```

Step 6 Press **Return**. The following prompt appears:

Example:

```
Router>
```

Step 7 Enter the enable command to enter enable mode. Configuration changes can be made only in enable mode:

Example:

```
Router> enable
```

The prompt changes to the privileged EXEC prompt:

Example:

```
Router#
```

Step 8 Enter the **show startup-config** command to display an enable password in the configuration file:

Example:

```
Router# show startup-config
```

What to do next

If you are recovering an enable password, do not perform the steps in the Reset the Password and Save Your Changes section. Instead, complete the password recovery process by performing the steps in the Reset the Configuration Register Value section.

If you are recovering an enable secret password, it is not displayed in the **show startup-config** command output. Complete the password recovery process by performing the steps in the Reset the Password and Save Your Changes section.

Reset the Password and Save Your Changes

To reset your password and save the changes, follow these steps:

Procedure

Step 1 Enter the **configure terminal** command to enter global configuration mode:

Example:

```
Router# configure terminal
```

Step 2 Enter the **enable secret** command to reset the enable secret password in the router:

Example:

```
Router(config)# enable secret
password
```

Step 3 Enter **exit** to exit global configuration mode:

Example:

```
Router(config)# exit
```

Step 4 Save your configuration changes:

Example:

```
Router# copy running-config startup-config
```

Reset the Configuration Register Value

To reset the configuration register value after you have recovered or reconfigured a password, follow these steps:

Procedure

Step 1 Enter the **configure terminal** command to enter global configuration mode:

Example:

```
Router# configure terminal
```

Step 2 Enter the **configure register** command and the original configuration register value that you recorded.

Example:

```
Router(config)# config-reg
value
```

Step 3 Enter **exit** to exit configuration mode:

Example:

```
Router(config)# exit
```

Note
To return to the configuration being used before you recovered the lost enable password, do not save the configuration changes before rebooting the router.

Step 4 Reboot the router, and enter the recovered password.

Configuring a Console Port Transport Map

This task describes how to configure a transport map for a console port interface on the router.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**transport-map type console** *transport-map-name* **Example:** `Router(config)# transport-map type console consolehandler`	Creates and names a transport map for handling console connections, and enters transport map configuration mode.
Step 4	**connection wait** [**allow** [**interruptible**] | **none** [**disconnect**]] **Example:** `Router(config-tmap)# connection wait none`	Specifies how a console connection will be handled using this transport map. • **allow interruptible**—The console connection waits for a Cisco IOS VTY line to become available, and also allows users to enter diagnostic mode by interrupting a console connection that is waiting for a Cisco IOS VTY line to become available. This is the default setting.

	Command or Action	Purpose
		Note Users can interrupt a waiting connection by entering **Ctrl-C** or **Ctrl-Shift-6**. • **none**—The console connection immediately enters diagnostic mode.

Step 5	(Optional) **banner** [**diagnostic** | **wait**] *banner-message*
	Example:
	`Router(config-tmap)# banner diagnostic X`
	`Enter TEXT message. End with the character 'X'.`
	`--Welcome to Diagnostic Mode--`
	`X`
	`Router(config-tmap)#`
		(Optional) Creates a banner message that will be seen by users entering diagnostic mode or waiting for the Cisco IOS VTY line because of the console transport map configuration.
		• **diagnostic**—Creates a banner message seen by users directed to diagnostic mode because of the console transport map configuration.
		Note Users can interrupt a waiting connection by entering **Ctrl-C** or **Ctrl-Shift-6**.
		• **wait**—Creates a banner message seen by users waiting for Cisco IOS VTY to become available.
		• *banner-message*—Banner message, which begins and ends with the same delimiting character.

Step 6	**exit** **Example:** `Router(config-tmap)# exit`	Exits transport map configuration mode to re-enter global configuration mode.
Step 7	**transport type console** *console-line-number* **input** *transport-map-name* **Example:** `Router(config)# transport type console 0 input consolehandler`	Applies the settings defined in the transport map to the console interface. The *transport-map-name* for this command must match the *transport-map-name* defined in the **transport-map type console** command.

Examples

The following example shows how to create a transport map to set console port access policies and attach to console port 0:

```
Router(config)# transport-map type console consolehandler
Router(config-tmap)# connection wait allow interruptible
Router(config-tmap)# banner diagnostic X
Enter TEXT message. End with the character 'X'.
```

```
--Welcome to diagnostic mode--
X
Router(config-tmap)# banner wait X
Enter TEXT message. End with the character 'X'.
Waiting for IOS vty line
X
Router(config-tmap)# exit
Router(config)# transport type console 0 input consolehandler
```

Viewing Console Port, SSH, and Telnet Handling Configurations

Use the following commands to view console port, SSH, and Telnet handling configurations:

- **show transport-map**
- **show platform software configuration access policy**

Use the **show transport-map** command to view transport map configurations.

show transport-map [**all** | **name** *transport-map-name* | **type** [**console**]]

This command can be used either in user EXEC mode or privileged EXEC mode.

Example

The following example shows transport maps that are configured on the router: console port (`consolehandler`):

```
Router# show transport-map all
Transport Map:
Name: consolehandler Type: Console Transport

Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI bshell banner:
Welcome to Diagnostic Mode



Router# show transport-map type console
Transport Map:
Name: consolehandler


REVIEW DRAFT - CISCO CONFIDENTIAL

Type: Console Transport

Connection:
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI Bshell banner:
Welcome to Diagnostic Mode



Router# show transport-map type persistent ssh
Transport Map:
Name: consolehandler Type: Console Transport

Connection:
```

```
Wait option: Wait Allow Interruptable Wait banner:

Waiting for the IOS CLI Bshell banner:
Welcome to Diagnostic Mode
```

Use the **show platform software configuration access policy** command to view the current configurations for handling the incoming console port, SSH, and Telnet connections. The output of this command provides the current wait policy for each type of connection (Telnet, SSH, and console), as well as information on the currently configured banners.

Unlike the **show transport-map** command, the **show platform software configuration access policy** command is available in diagnostic mode so that it can be entered in scenarios where you need transport map configuration information, but cannot access the Cisco IOS CLI.

Example

The following example shows the **show platform software configuration access policy** command.

```
Router# show platform software configuration access policy
The current access-policies

Method : telnet
Rule : wait with interrupt Shell banner:
Welcome to Diagnostic Mode

Wait banner :
Waiting for IOS Process

Method : ssh Rule : wait Shell banner: Wait banner :

Method : console
Rule : wait with interrupt Shell banner:
Wait banner :
```

Using the factory reset Commands

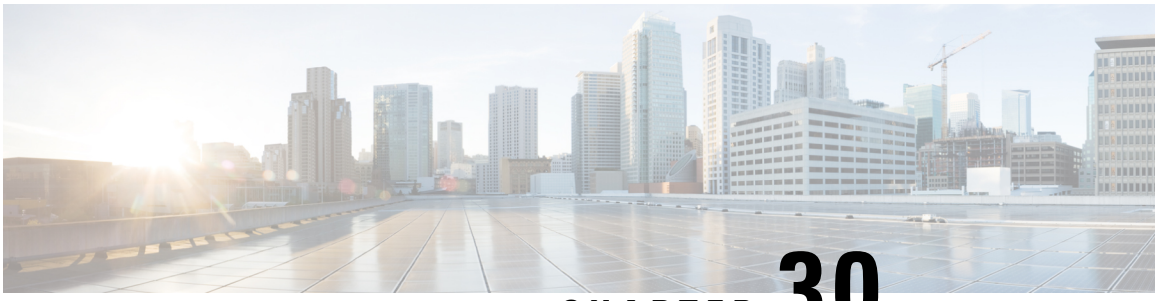
The **factory reset** commands are used to remove all the customer specific data on a router/switch that has been added. The data can be configuration, log files, boot variables, core files, and so on.

The **factory-reset all** command erases the bootflash, nvram, rommon variables, licenses, and logs.

```
Router#factory-reset all
The factory reset operation is irreversible for all operations. Are you sure? [confirm]
*Enter*

*May 12 09:55:45.831: %SYS-5-RELOAD: Reload requested by Exec. Reload Reason: Factory Reset.

***Return to ROMMON Prompt
```



CHAPTER 30

System Messages

This chapter contains the following sections:

- Information About Process Management, on page 321
- How to Find Error Message Details, on page 321

Information About Process Management

You can access system messages by logging in to the console through Telnet protocol and monitoring your system components remotely from any workstation that supports the Telnet protocol.

Starting and monitoring software is referred to as process management. The process management infrastructure for a router is platform independent, and error messages are consistent across platforms running on Cisco IOS XE. You do not have to be directly involved in process management, but we recommend that you read the system messages that refer to process failures and other issues.

How to Find Error Message Details

To show further details about a process management or a syslog error message, enter the error message into the Error Message Decoder tool at: https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi.

For example, enter the message `%PMAN-0-PROCESS_NOTIFICATION` into the tool to view an explanation of the error message and the recommended action to be taken.

The following are examples of the description and the recommended action displayed by the Error Message Decoder tool for some of the error messages.

Error Message: `%PMAN-0-PROCESS_NOTIFICATION : The process lifecycle notification component failed because [chars]`

Explanation	Recommended Action

The process lifecycle notification component failed, preventing proper detection of a process start and stop. This problem is likely the result of a software defect in the software subpackage.	Note the time of the message and investigate the kernel error message logs to learn more about the problem and see if it is correctable. If the problem cannot be corrected or the logs are not helpful, copy the error message exactly as it appears on the console along with the output of the **show tech-support** command and provide the gathered information to a Cisco technical support representative.

Error Message: `%PMAN-0-PROCFAILCRIT A critical process [chars] has failed (rc [dec])`

Explanation	Recommended Action
A process important to the functioning of the router has failed.	Note the time of the message and investigate the error message logs to learn more about the problem. If the problem persists, copy the message exactly as it appears on the console or in the system log. Research and attempt to resolve the issue using the tools and utilities provided at: http://www.cisco.com/tac. With some messages, these tools and utilities will supply clarifying information. Search for resolved software issues using the Bug Search Tool at: http://www.cisco.com/cisco/psn/bssprt/bss. If you still require assistance, open a case with the Technical Assistance Center at: http://tools.cisco.com/ServiceRequestTool/create/, or contact your Cisco technical support representative and provide the representative with the information you have gathered. Attach the following information to your case in nonzipped, plain-text (.txt) format: the output of the **show logging** and **show tech-support** commands and your pertinent troubleshooting logs.

Error Message: `%PMAN-3-PROCFAILOPT An optional process [chars] has failed (rc [dec])`

Explanation	Recommended Action

A process that does not affect the forwarding of traffic has failed.	Note the time of the message and investigate the kernel error message logs to learn more about the problem. Although traffic will still be forwarded after receiving this message, certain functions on the router may be disabled because of this message and the error should be investigated. If the logs are not helpful or indicate a problem you cannot correct, copy the message exactly as it appears on the console or in the system log. Research and attempt to resolve the issue using the tools and utilities provided at http://www.cisco.com/tac. With some messages, these tools and utilities will supply clarifying information. Search for resolved software issues using the Bug Search Tool at: http://www.cisco.com/cisco/psn/bssprt/bss. If you still require assistance, open a case with the Technical Assistance Center at: http://tools.cisco.com/ServiceRequestTool/create/, or contact your Cisco technical support representative and provide the representative with the information you have gathered. Attach the following information to your case in nonzipped, plain-text (.txt) format: the output of the **show logging** and **show tech-support** commands and your pertinent troubleshooting logs.

Error Message: `%PMAN-3-PROCFAIL The process [chars] has failed (rc [dec])`

Explanation	Recommended Action
The process has failed as the result of an error.	This message will appear with other messages related to the process. Check the other messages to determine the reason for the failures and see if corrective action can be taken. If the problem persists, copy the message exactly as it appears on the console or in the system log. Research and attempt to resolve the issue using the tools and utilities provided at: http://www.cisco.com/tac. With some messages, these tools and utilities will supply clarifying information. Search for resolved software issues using the Bug Search Tool at: http://www.cisco.com/cisco/psn/bssprt/bss. If you still require assistance, open a case with the Technical Assistance Center at: http://tools.cisco.com/ServiceRequestTool/create/, or contact your Cisco technical support representative and provide the representative with the information you have gathered. Attach the following information to your case in nonzipped, plain-text (.txt) format: the output of the **show logging** and **show tech-support** commands and your pertinent troubleshooting logs.

Error Message: `%PMAN-3-PROCFAIL_IGNORE [chars] process exits and failures are being ignored due to debug settings. Normal router functionality will be affected. Critical router functions like RP switchover, router reload, FRU resets, etc. may not function properly.`

Explanation	Recommended Action
A process failure is being ignored due to the user-configured debug settings.	If this behavior is desired and the debug settings are set according to a user's preference, no action is needed. If the appearance of this message is viewed as a problem, change the debug settings. The router is not expected to behave normally with this debug setting. Functionalities such as SSO switchover, router reloads, FRU resets, and so on will be affected. This setting should only be used in a debug scenario. It is not normal to run the router with this setting.

Error Message: `%PMAN-3-PROCHOLDDOWN The process [chars] has been helddown (rc [dec])`

Explanation	Recommended Action
The process was restarted too many times with repeated failures and has been placed in the hold-down state.	This message will appear with other messages related to the process. Check the other messages to determine the reason for the failures and see if corrective action can be taken. If the problem persists, copy the message exactly as it appears on the console or in the system log. Research and attempt to resolve the issue using the tools and utilities provided at: http://www.cisco.com/tac. With some messages, these tools and utilities will supply clarifying information. Search for resolved software issues using the Bug Search Tool at: http://www.cisco.com/cisco/psn/bssprt/bss. If you still require assistance, open a case with the Technical Assistance Center at: http://tools.cisco.com/ServiceRequestTool/create/, or contact your Cisco technical support representative and provide the representative with the information you have gathered. Attach the following information to your case in nonzipped, plain-text (.txt) format: the output of the **show logging** and **show tech-support** commands and your pertinent troubleshooting logs.

Error Message: `%PMAN-3-RELOAD_RP_SB_NOT_READY : Reloading: [chars]`

Explanation	Recommended Action
The route processor is being reloaded because there is no ready standby instance.	Ensure that the reload is not due to an error condition.

Error Message: `%PMAN-3-RELOAD_RP : Reloading: [chars]`

Explanation	Recommended Action

The RP is being reloaded.	Ensure that the reload is not due to an error condition. If it is due to an error condition, collect information requested by the other log messages.

Error Message: `%PMAN-3-RELOAD_SYSTEM : Reloading: [chars]`

Explanation	Recommended Action
The system is being reloaded.	Ensure that the reload is not due to an error condition. If it is due to an error condition, collect information requested by the other log messages.

Error Message: `%PMAN-3-PROC_BAD_EXECUTABLE : Bad executable or permission problem with process [chars]`

Explanation	Recommended Action
The executable file used for the process is bad or has permission problem.	Ensure that the named executable is replaced with the correct executable.

Error Message: `%PMAN-3-PROC_BAD_COMMAND:Non-existent executable or bad library used for process <process name>`

Explanation	Recommended Action
The executable file used for the process is missing, or a dependent library is bad.	Ensure that the named executable is present and the dependent libraries are good.

Error Message: `%PMAN-3-PROC_EMPTY_EXEC_FILE : Empty executable used for process [chars]`

Explanation	Recommended Action
The executable file used for the process is empty.	Ensure that the named executable is non-zero in size.

Error Message: `%PMAN-5-EXITACTION : Process manager is exiting: [chars]`

Explanation	Recommended Action
The process manager is exiting.	Ensure that the process manager is not exiting due to an error condition. If it is due to an error condition, collect information requested by the other log messages.

Error Message: `%PMAN-6-PROCSHUT : The process [chars] has shutdown`

Explanation	Recommended Action
The process has gracefully shut down.	No user action is necessary. This message is provided for informational purposes only.

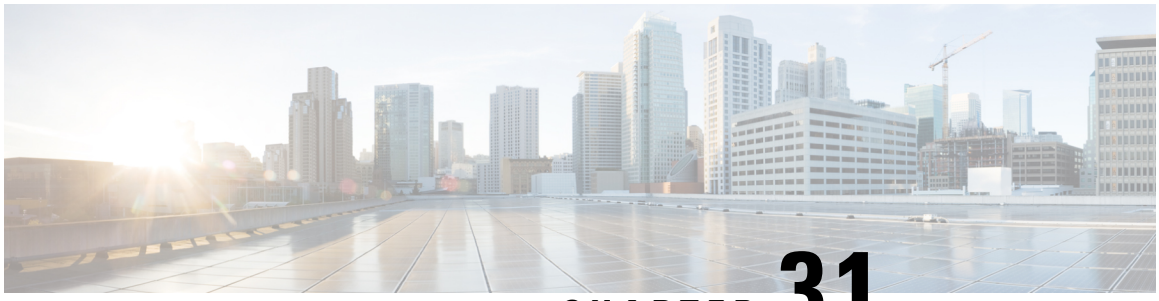
Error Message: `%PMAN-6-PROCSTART : The process [chars] has started`

Explanation	Recommended Action

The process has launched and is operating properly.	No user action is necessary. This message is provided for informational purposes only.

Error Message: `%PMAN-6-PROCSTATELESS : The process [chars] is restarting stateless`

Explanation	Recommended Action
The process has requested a stateless restart.	No user action is necessary. This message is provided for informational purposes only.



CHAPTER 31

Process Health Monitoring

This chapter describes how to manage and monitor the health of various components of your router. It contains the following sections:

- Monitoring Control Plane Resources, on page 327
- Monitoring Hardware Using Alarms, on page 332
- Approaches for Monitoring Hardware Alarms, on page 333

Monitoring Control Plane Resources

The following sections explain the details of memory and CPU monitoring from the perspective of the Cisco IOS process and the overall control plane:

- Avoiding Problems Through Regular Monitoring, on page 327
- Cisco IOS XE Process Resources, on page 327
- Overall Control Plane Resources, on page 331

Avoiding Problems Through Regular Monitoring

Processes should provide monitoring and notification of their status/health to ensure correct operation. When a process fails, a syslog error message is displayed and either the process is restarted or the router is rebooted. A syslog error message is displayed when a monitor detects that a process is stuck or has crashed. If the process can be restarted, it is restarted; else, the router is restarted.

Monitoring system resources enables you to detect potential problems before they occur, thus avoiding outages. It also establishes a baseline for a normal system load. You can use this information as a basis for comparison, when you upgrade hardware or software to see if the upgrade has affected resource usage.

Cisco IOS XE Process Resources

You can view CPU utilization statistics on active processes and see the amount of memory being used in these processes using the **show memory** command and the **show process cpu** command. These commands provide a representation of memory and CPU utilization from the perspective of only the Cisco IOS process; they do not include information for resources on the entire platform. When the **show memory** command is used in a system with 4 GB RAM running a single Cisco IOS process, the following memory usage is displayed:

```
Router# show memory
Tracekey : 1#33e0077971693714bd2b0bc347d77489
Address Bytes Prev Next Ref PrevF NextF what Alloc PC

Head Total(b) Used(b) Free(b) Lowest(b) Largest(b)
Processor 7F68ECD010 728952276 281540188 447412088 445683380 234766720
lsmpi_io 7F6852A1A8 6295128 6294304 824 824 412
Dynamic heap limit(MB) 200 Use(MB) 0



Processor memory

Address Bytes Prev Next Ref PrevF NextF what Alloc PC
7F68ECD010 0000000568 00000000 7F68ECD2A0 001 -------- -------- *Init* :400000+60E37C4
7F68ECD2A0 0000032776 7F68ECD010 7F68ED5300 001 -------- -------- Managed Chunk Q
:400000+60D12A8
7F68ED5300 0000000056 7F68ECD2A0 7F68ED5390 001 -------- -------- *Init* :400000+3B0C610
7F68ED5390 0000012808 7F68ED5300 7F68ED85F0 001 -------- -------- *Init* :400000+B8A5D64
Address Bytes Prev Next Ref PrevF NextF what Alloc PC
7F68ED85F0 0000032776 7F68ED5390 7F68EE0650 001 -------- -------- List Elements
:400000+60A4A9C
7F68EE0650 0000032776 7F68ED85F0 7F68EE86B0 001 -------- -------- List Headers :400000+60A4AD8
7F68EE86B0 0000032776 7F68EE0650 7F68EF0710 001 -------- -------- IOSXE Process S
:400000+11924CC
7F68EF0710 0000032776 7F68EE86B0 7F68EF8770 001 -------- -------- IOSXE Queue Pro
:400000+1192510
7F68EF8770 0000065544 7F68EF0710 7F68F087D0 001 -------- -------- IOSXE Queue Bal
:400000+1192554
7F68F087D0 0000000328 7F68EF8770 7F68F08970 001 -------- -------- *Init* :400000+B89E1D8
7F68F08970 0000000328 7F68F087D0 7F68F08B10 001 -------- -------- *Init* :400000+B89E1D8
7F68F08B10 0000000328 7F68F08970 7F68F08CB0 001 -------- -------- *Init* :400000+B89E1D8
7F68F08CB0 0000000360 7F68F08B10 7F68F08E70 001 -------- -------- Process Events
:400000+60F9CD4
7F68F08E70 0000000056 7F68F08CB0 7F68F08F00 001 -------- -------- SDB String :400000+605981C
7F68F08F00 0000000080 7F68F08E70 7F68F08FA8 001 -------- -------- Init :400000+60599E4
Address Bytes Prev Next Ref PrevF NextF what Alloc PC
7F68F08FA8 0000036872 7F68F08F00 7F68F12008 001 -------- -------- *Init* :400000+11891E8
7F68F12008 0000010008 7F68F08FA8 7F68F14778 001 -------- -------- Platform VM Pag
:400000+11AD244
7F68F14778 0000002008 7F68F12008 7F68F14FA8 001 -------- -------- *Init*
iosd_crb_sparrow_unix:7F8EB59000+5CC1C
7F68F14FA8 0000200712 7F68F14778 7F68F46008 001 -------- -------- Interrupt Stack
:400000+11891E8
7F68F46008 0000003008 7F68F14FA8 7F68F46C20 001 -------- -------- Watched Semapho
:400000+60FE448
7F68F46C20 0000000328 7F68F46008 7F68F46DC0 001 -------- -------- *Init* :400000+B89E1D8
7F68F46DC0 0000000096 7F68F46C20 7F68F46E78 001 -------- -------- Init :400000+60599E4
7F68F46E78 0000000216 7F68F46DC0 7F68F46FA8 001 -------- -------- *Init* :400000+60ED228
7F68F46FA8 0000036872 7F68F46E78 7F68F50008 001 -------- -------- *Init* :400000+11891E8
7F68F50008 0000000896 7F68F46FA8 7F68F503E0 001 -------- -------- Watched Message
:400000+60FE4A8
7F68F503E0 0000002008 7F68F50008 7F68F50C10 001 -------- -------- Watcher Message
:400000+60FE4D8
Address Bytes Prev Next Ref PrevF NextF what Alloc PC
7F68F50C10 0000000360 7F68F503E0 7F68F50DD0 001 -------- -------- Process Events
:400000+60F9CD4
7F68F50DD0 0000000184 7F68F50C10 7F68F50EE0 001 -------- -------- *Init* :400000+60ED918
7F68F50EE0 0000000112 7F68F50DD0 7F68F50FA8 001 -------- -------- *Init* :400000+60B57CC
7F68F50FA8 0000036872 7F68F50EE0 7F68F5A008 001 -------- -------- *Init* :400000+11891E8
7F68F5A008 0000002336 7F68F50FA8 7F68F5A980 001 -------- -------- Process Array
:400000+6102A4C
7F68F5A980 0000000184 7F68F5A008 7F68F5AA90 001 -------- -------- *Init* :400000+60ED918
```

```
7F68F5AA90 0000000184 7F68F5A980 7F68F5ABA0 001 -------- -------- *Init* :400000+60ED918
7F68F5ABA0 0000000184 7F68F5AA90 7F68F5ACB0 001 -------- -------- *Init* :400000+60ED918
7F68F5ACB0 0000000184 7F68F5ABA0 7F68F5ADC0 001 -------- -------- *Init* :400000+60ED918
7F68F5ADC0 0000000184 7F68F5ACB0 7F68F5AED0 001 -------- -------- *Init* :400000+60ED918
```

The **show process cpu** command displays Cisco IOS CPU utilization average:

```
Router# show process cpu
CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
1 0 17 0 0.00% 0.00% 0.00% 0 Chunk Manager
2 552 1205 458 0.00% 0.00% 0.00% 0 Load Meter
3 0 1 0 0.00% 0.00% 0.00% 0 PKI Trustpool
4 0 1 0 0.00% 0.00% 0.00% 0 Retransmission o
5 0 1 0 0.00% 0.00% 0.00% 0 IPC ISSU Dispatc
6 36 13 2769 0.00% 0.00% 0.00% 0 RF Slave Main Th
7 0 1 0 0.00% 0.00% 0.00% 0 EDDRI_MAIN
8 0 1 0 0.00% 0.00% 0.00% 0 RO Notify Timers
9 4052 920 4404 0.23% 0.09% 0.06% 0 Check heaps
10 12 101 118 0.00% 0.00% 0.00% 0 Pool Manager
11 0 1 0 0.00% 0.00% 0.00% 0 DiscardQ Backgro
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
12 0 2 0 0.00% 0.00% 0.00% 0 Timers
13 0 163 0 0.00% 0.00% 0.00% 0 WATCH_AFS
14 0 2 0 0.00% 0.00% 0.00% 0 ATM AutoVC Perio
15 0 2 0 0.00% 0.00% 0.00% 0 ATM VC Auto Crea
16 76 3024 25 0.00% 0.00% 0.00% 0 IOSXE heartbeat
17 0 13 0 0.00% 0.00% 0.00% 0 DB Lock Manager
18 0 1 0 0.00% 0.00% 0.00% 0 DB Notification
19 0 1 0 0.00% 0.00% 0.00% 0 IPC Apps Task
20 0 1 0 0.00% 0.00% 0.00% 0 ifIndex Receive
21 36 1210 29 0.00% 0.00% 0.00% 0 IPC Event Notifi
22 72 5904 12 0.00% 0.00% 0.00% 0 IPC Mcast Pendin
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
23 0 1 0 0.00% 0.00% 0.00% 0 Platform appsess
24 0 101 0 0.00% 0.00% 0.00% 0 IPC Dynamic Cach
25 16 1210 13 0.00% 0.00% 0.00% 0 IPC Service NonC
26 0 1 0 0.00% 0.00% 0.00% 0 IPC Zone Manager
27 64 5904 10 0.00% 0.00% 0.00% 0 IPC Periodic Tim
28 76 5904 12 0.00% 0.00% 0.00% 0 IPC Deferred Por
29 0 1 0 0.00% 0.00% 0.00% 0 IPC Process leve
30 0 1 0 0.00% 0.00% 0.00% 0 IPC Seat Manager
31 8 346 23 0.00% 0.00% 0.00% 0 IPC Check Queue
32 0 1 0 0.00% 0.00% 0.00% 0 IPC Seat RX Cont
33 0 1 0 0.00% 0.00% 0.00% 0 IPC Seat TX Cont
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
34 48 606 79 0.00% 0.00% 0.00% 0 IPC Keep Alive M
35 28 1210 23 0.00% 0.00% 0.00% 0 IPC Loadometer
36 0 1 0 0.00% 0.00% 0.00% 0 IPC Session Deta
37 0 1 0 0.00% 0.00% 0.00% 0 SENSOR-MGR event
38 4 606 6 0.00% 0.00% 0.00% 0 Compute SRP rate
39 0 1 0 0.00% 0.00% 0.00% 0 MEMLEAK PROCESS
40 0 1 0 0.00% 0.00% 0.00% 0 ARP Input
41 112 6331 17 0.00% 0.00% 0.00% 0 ARP Background
42 0 2 0 0.00% 0.00% 0.00% 0 ATM Idle Timer
43 0 1 0 0.00% 0.00% 0.00% 0 ATM ASYNC PROC
44 0 1 0 0.00% 0.00% 0.00% 0 CEF MIB API
--More--
...
show process cpu platform sorted

CPU utilization for five seconds: 11%, one minute: 12%, five minutes: 12%
```

```
Core 0: CPU utilization for five seconds: 1%, one minute: 3%, five minutes: 3%
Core 1: CPU utilization for five seconds: 1%, one minute: 3%, five minutes: 3%
Core 2: CPU utilization for five seconds: 1%, one minute: 1%, five minutes: 1%
Core 3: CPU utilization for five seconds: 42%, one minute: 42%, five minutes: 42%
Pid PPid 5Sec 1Min 5Min Status Size Name
------------------------------------------------------------------------------
18246 17700 34% 34% 34% S 272500 qfp-ucode-sparr
18297 16477 1% 1% 1% S 165768 fman_fp_image
9992 9121 1% 1% 1% S 743608 linux_iosd-imag
27122 26048 0% 0% 0% S 8460 nginx
26048 25864 0% 0% 0% S 19252 nginx
25928 1 0% 0% 0% S 2960 rotee
25864 1 0% 0% 0% S 3532 pman.sh
24212 2 0% 0% 0% S 0 kworker/u8:0
19648 8282 0% 0% 0% S 220 sleep
19635 10903 0% 0% 0% S 212 sleep
18121 17675 0% 0% 0% S 10968 ngiolite
17979 1 0% 0% 0% S 1660 rotee
17863 2 0% 0% 0% S 0 kworker/1:0
17859 1 0% 0% 0% S 2836 rotee
17737 17095 0% 0% 0% S 56828 iomd
17700 13380 0% 0% 0% S 3556 pman.sh
17675 12798 0% 0% 0% S 3524 pman.sh
17518 16854 0% 0% 0% S 15024 hman
17312 1 0% 0% 0% S 2828 rotee
17095 12798 0% 0% 0% S 3568 pman.sh
17085 1 0% 0% 0% S 2876 rotee
16942 2 0% 0% 0% S 0 kworker/0:1
16892 14768 0% 0% 0% S 108952 cpp_cp_svr
16854 13380 0% 0% 0% S 3568 pman.sh
16716 1 0% 0% 0% S 2996 rotee
16664 15963 0% 0% 0% S 51096 cpp_sp_svr
16477 13380 0% 0% 0% S 3540 pman.sh
16326 15536 0% 0% 0% S 39852 cpp_ha_top_leve
16270 1 0% 0% 0% S 2972 rotee
15963 13380 0% 0% 0% S 3528 pman.sh
15779 15163 0% 0% 0% S 55208 cpp_driver
15730 1 0% 0% 0% S 1640 rotee
15536 13380 0% 0% 0% S 3528 pman.sh
15412 1 0% 0% 0% S 1716 rotee
15274 14681 0% 0% 0% S 15004 hman
15163 13380 0% 0% 0% S 3624 pman.sh
15083 14361 0% 0% 0% S 26792 cman_fp
15057 1 0% 0% 0% S 1660 rotee
14891 1 0% 0% 0% S 2868 rotee
14768 13380 0% 0% 0% S 3568 pman.sh
14722 14127 0% 0% 0% S 27536 cmcc
14717 14108 0% 0% 0% S 15220 btman
14681 12798 0% 0% 0% S 3572 pman.sh
14627 1 0% 0% 0% S 2996 rotee
14361 13380 0% 0% 0% S 3596 pman.sh
14338 1 0% 0% 0% S 2984 rotee
14314 1 0% 0% 0% S 2824 rotee
14155 13577 0% 0% 0% S 15128 btman
14127 12798 0% 0% 0% S 3612 pman.sh
14108 13380 0% 0% 0% S 3572 pman.sh
13813 13380 0% 0% 0% S 252 inotifywait
--More--
```

Overall Control Plane Resources

Control plane memory and CPU utilization on each control processor allows you to keep a tab on the overall control plane resources. You can use the **show platform software status control-processor brief** command (summary view) or the **show platform software status control-processor** command (detailed view) to view control plane memory and CPU utilization information.

All control processors should show status, Healthy. Other possible status values are Warning and Critical. Warning indicates that the router is operational, but that the operating level should be reviewed. Critical implies that the router is nearing failure.

If you see a Warning or Critical status, take the following actions:

- Reduce the static and dynamic loads on the system by reducing the number of elements in the configuration or by limiting the capacity for dynamic services.
- Reduce the number of routes and adjacencies, limit the number of ACLs and other rules, reduce the number of VLANs, and so on.

The following sections describe the fields in the **show platform software status control-processor** command output.

Load Average

Load average represents the process queue or process contention for CPU resources. For example, on a single-core processor, an instantaneous load of 7 would mean that seven processes are ready to run, one of which is currently running. On a dual-core processor, a load of 7 would mean that seven processes are ready to run, two of which are currently running.

Memory Utilization

Memory utilization is represented by the following fields:

- Total—Total system memory
- Used—Consumed memory
- Free—Available memory
- Committed—Virtual memory committed to processes

CPU Utilization

CPU utilization is an indication of the percentage of time the CPU is busy, and is represented by the following fields:

- CPU—Allocated processor
- User—Non-Linux kernel processes
- System—Linux kernel process
- Nice—Low-priority processes
- Idle—Percentage of time the CPU was inactive
- IRQ—Interrupts

- SIRQ—System Interrupts
- IOwait—Percentage of time CPU was waiting for I/O

show platform software status control-processor Command

The following are some examples of using the **show platform software status control-processor** command:

```
Router# show platform software status control-processor
RP0: online, statistics updated 4 seconds ago
Load Average: healthy
1-Min: 0.29, status: healthy, under 5.00
5-Min: 0.51, status: healthy, under 5.00
15-Min: 0.54, status: healthy, under 5.00
Memory (kb): healthy
Total: 4038072
Used: 2872136 (71%), status: healthy
Free: 1165936 (29%)
Committed: 2347228 (58%), under 90%
Per-core Statistics
CPU0: CPU Utilization (percentage of time spent)
User: 1.00, System: 0.70, Nice: 0.00, Idle: 97.88
IRQ: 0.30, SIRQ: 0.10, IOwait: 0.00
CPU1: CPU Utilization (percentage of time spent)
User: 0.70, System: 0.30, Nice: 0.00, Idle: 98.48
IRQ: 0.30, SIRQ: 0.20, IOwait: 0.00
CPU2: CPU Utilization (percentage of time spent)
User: 0.20, System: 1.11, Nice: 0.00, Idle: 98.27
IRQ: 0.40, SIRQ: 0.00, IOwait: 0.00
CPU3: CPU Utilization (percentage of time spent)
User: 8.23, System: 24.37, Nice: 0.00, Idle: 58.00
IRQ: 9.26, SIRQ: 0.11, IOwait: 0.00
```

```
Router# show platform software status control-processor brief
Load Average
Slot Status 1-Min 5-Min 15-Min
RP0 Healthy 0.28 0.46 0.52

Memory (kB)
Slot Status Total Used (Pct) Free (Pct) Committed (Pct)
RP0 Healthy 4038072 2872672 (71%) 1165400 (29%) 2349820 (58%)

CPU Utilization
Slot CPU User System Nice Idle IRQ SIRQ IOwait
RP0 0 0.70 0.20 0.00 98.58 0.30 0.20 0.00
1 1.10 0.90 0.00 97.59 0.30 0.10 0.00
2 0.40 1.31 0.00 97.87 0.40 0.00 0.00
3 8.00 26.55 0.00 56.33 8.99 0.11 0.00
```

Monitoring Hardware Using Alarms

This section contains the following:

Router Design and Monitoring Hardware

The router sends alarm notifications when problems are detected, allowing you to monitor the network remotely. You do not need to use **show** commands to poll devices on a routine basis; however, you can perform onsite monitoring if you choose.

BootFlash Disk Monitoring

The bootflash disk must have enough free space to store two core dumps. This condition is monitored, and if the bootflash disk is too small to store two core dumps, a syslog alarm is generated, as shown in the following example:

```
Oct  6 14:10:56.292: %FLASH_CHECK-3-DISK_QUOTA: R0/0: flash_check: Flash disk quota exceeded

[free space is 1429020 kB] - Please clean up files on bootflash.
```

Approaches for Monitoring Hardware Alarms

This section contains the following:

Viewing the Console or Syslog for Alarm Messages

The network administrator can monitor alarm messages by reviewing alarm messages sent to the system console or to a system message log (syslog).

Enabling the logging alarm Command

The **logging alarm** command must be enabled for the system to send alarm messages to a logging device, such as the console or a syslog. This command is not enabled by default.

You can specify the severity level of the alarms to be logged. All the alarms at and above the specified threshold generate alarm messages. For example, the following command sends only critical alarm messages to logging devices:

```
Router(config)# logging alarm critical
```

If alarm severity is not specified, alarm messages for all severity levels are sent to logging devices.

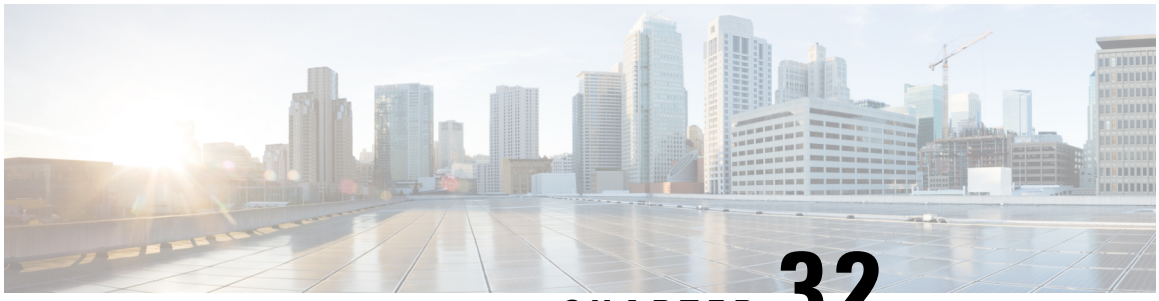
Network Management System Alerts a Network Administrator when an Alarm is Reported Through SNMP

The SNMP is an application-layer protocol that provides a standardized framework and a common language used for monitoring and managing devices in a network.

SNMP provides notification of faults, alarms, and conditions that might affect services. It allows a network administrator to access router information through a network management system (NMS) instead of reviewing logs, polling devices, or reviewing log reports.

To use SNMP to get alarm notification, use the following MIBs:

- ENTITY-MIB, RFC4133 (required for the CISCO-ENTITY-ALARM-MIB, ENTITY-STATE-MIB and CISCO-ENTITY-SENSOR-MIB to work)
- CISCO-ENTITY-ALARM-MIB
- ENTITY-STATE-MIB
- CISCO-ENTITY-SENSOR-MIB (for transceiver environmental alarm information, which is not provided through the CISCO-ENTITY-ALARM-MIB)



CHAPTER 32

Environmental Monitoring

This chapter contains the following sections:

- Environmental Monitoring, on page 335
- Environmental Monitoring, on page 335
- Environmental Monitoring and Reporting Functions, on page 336
- Environmental Monitoring Functions, on page 336
- Environmental Reporting Functions, on page 337
- SNMP Polling of Temperature OID, on page 343
- Technical Assistance, on page 344
- Additional References, on page 344

Environmental Monitoring

This chapter contains the following sections:

Environmental Monitoring

The router provides a robust environment-monitoring system with several sensors that monitor the system temperatures. The following are some of the key functions of the environmental monitoring system:

- Monitoring temperature of CPUs and Motherboard
- Recording abnormal events and generating notifications
- Monitoring Simple Network Management Protocol (SNMP) traps
- Generating and collecting Onboard Failure Logging (OBFL) data
- Sending call home event notifications
- Logging system error messages
- Displaying present settings and status

Environmental Monitoring and Reporting Functions

Monitoring and reporting functions allow you to maintain normal system operation by identifying and resolving adverse conditions prior to loss of operation.

- Environmental Monitoring Functions, on page 336
- Environmental Reporting Functions, on page 337

Environmental Monitoring Functions

Environmental monitoring functions use sensors to monitor the temperature of the cooling air as it moves through the chassis.

The router is expected to meet the following environmental operating conditions

- Non-operating Temperature: -40F to +185F (-40C to +85C)
- Non-operating Humidity: 5 to 95% relative humidity (non-condensing)
- Operating Temperature:

  -40F to +185F (-40C to +85C) as measured at the center of the top surface of the –CON SKU thermal plate.

  Temperature range of a completed solution depends on the enclosure thermal design characteristics used by the integrator.

  If –NCP SKU is used, integrator is responsible for designing a thermal solution that meets the component level requirements provided in this document.
- Operating Humidity: 95% +/- 5% RH
- Operating Altitude: -15,000ft (4,572m)

The following table displays the levels of status conditions used by the environmental monitoring system.

Table 25: Levels of Status Conditions Used by the Environmental Monitoring System

Status Level	Description
Normal	All monitored parameters are within normal tolerance.
Warning	The system has exceeded a specified threshold. The system continues to operate, but operator action is recommended to bring the system back to a normal state.
Critical	An out-of-tolerance temperature or voltage condition exists. Although the system continues to operate, it is approaching shutdown. Immediate operator action is required.

The environmental monitoring system sends system messages to the console, for example, when the conditions described here are met:

Temperature and Voltage Exceed Max/Min Thresholds

The following example shows the warning messages indicating the maximum and minimum thresholds of the temperature or voltage:

```
Warnings :
--------
For all the temperature sensors (name starting with “Temp:”) above,
the critical warning threshold is 100C (100C and higher)
the warning threshold is 80C (range from 80C to 99C)
the low warning threshold is 1C (range from -inf to 1C).

For all voltage sensors (names starting with "V:"),
the high warning threshold starts at that voltage +10%. (voltage + 10% is warning)
the low warning threshold starts at the voltage -10%. (voltage - 10% is warning)
```

Environmental Reporting Functions

You can retrieve and display environmental status reports using the following commands:

- **show diag all eeprom**
- **show environment**
- **show environment all**
- **show inventory**
- **show platform**
- **show platform diag**
- **show platform software status control-processor**
- **show diag slot R0 eeprom detail**
- **show version**
- **show power**

These commands show the current values of parameters such as temperature and voltage.

The environmental monitoring system updates the values of these parameters every 60 seconds. Brief examples of these commands are shown below:

show diag all eeprom: Example

```
Router#show diag all eeprom
MIDPLANE EEPROM data:

Product Identifier (PID) : ESR-6300-NCP-K9
Version Identifier (VID) : V00
PCB Serial Number : FOC23032UUN
Top Assy. Revision : 01
Hardware Revision : 0.2
Asset ID :
CLEI Code : UNASSIGNED
External PoE Module POE0 EEPROM data is not initialized
```

```
Internal PoE is not present

Slot R0 EEPROM data:

Product Identifier (PID) : ESR-6300-NCP-K9
Version Identifier (VID) : V00
PCB Serial Number : FOC23032UUN
Top Assy. Revision : 01
Hardware Revision : 0.2
CLEI Code : UNASSIGNED
Slot F0 EEPROM data:

Product Identifier (PID) : ESR-6300-NCP-K9
Version Identifier (VID) : V00
PCB Serial Number : FOC23032UUN
Top Assy. Revision : 01
Hardware Revision : 0.2
CLEI Code : UNASSIGNED
Slot 0 EEPROM data:

Product Identifier (PID) : ESR-6300-NCP-K9
Version Identifier (VID) : V00
PCB Serial Number : FOC23032UUN
Top Assy. Revision : 01
Hardware Revision : 0.2
CLEI Code : UNASSIGNED
SPA EEPROM data for subslot 0/0:

Product Identifier (PID) : ESR-6300-2x1GE
Version Identifier (VID) : V01
PCB Serial Number :
Top Assy. Part Number : 68-2236-01
Top Assy. Revision : A0
Hardware Revision : 2.2
CLEI Code : CNUIAHSAAA
SPA EEPROM data for subslot 0/1:

Product Identifier (PID) : ESR-6300-ES-4
Version Identifier (VID) : V01
PCB Serial Number :
Top Assy. Part Number : 68-2236-01
Top Assy. Revision : A0
Hardware Revision : 2.2
CLEI Code : CNUIAHSAAA
SPA EEPROM data for subslot 0/2 is not available

SPA EEPROM data for subslot 0/3 is not available

SPA EEPROM data for subslot 0/4 is not available

SPA EEPROM data for subslot 0/5 is not available
```

show environment: Example

```
Router#show environment
Number of Critical alarms: 0
Number of Major alarms: 0
Number of Minor alarms: 0

Slot Sensor Current State Reading Threshold(Minor,Major,Critical,Shutdown)
```

```
---------- -------------- --------------- ------------ -------------------------------------
R0 Temp: LM75BXXX Normal 45 Celsius (75 ,80 ,90 ,na )(Celsius)
```

show environment all: Example

```
Router#show environment all
Sensor List: Environmental Monitoring
Sensor Location State Reading
Temp: LM75BXXX R0 Normal 45 Celsius
```

show inventory: Example

```
Router#show inventory
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INFO: Please use "show license UDI" to get serial number for licensing.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

NAME: "Chassis", DESCR: "Embedded Services Router with 2 Gigabit Ethernet routed ports,4
Gigabit Ethernet switched ports, no cooling plate"
PID: ESR-6300-NCP-K9 , VID: V00 , SN: FOC23032UUN

NAME: "Power Supply Module 0", DESCR: "External Power Supply Module"
PID: PWR-12V , VID: , SN:

NAME: "module 0", DESCR: "Cisco ESR-6300-NCP-K9 Fixed and pluggable Interface Module
controller"
PID: ESR-6300-NCP-K9 , VID: , SN:

NAME: "NIM subslot 0/1", DESCR: "ESR-6300-ES-4"
PID: ESR-6300-ES-4 , VID: V01 , SN:

NAME: "NIM subslot 0/0", DESCR: "Front Panel 2 port Gigabitethernet Module"
PID: ESR-6300-2x1GE , VID: V01 , SN:

NAME: "module R0", DESCR: "Cisco ESR-6300-NCP-K9 Route Processor"
PID: ESR-6300-NCP-K9 , VID: V00 , SN: FOC23032UUN

NAME: "module F0", DESCR: "Cisco ESR-6300-NCP-K9 Forwarding Processor"
PID: ESR-6300-NCP-K9 , VID: , SN:
```

show platform: Example

```
Router#show platform
Chassis type: ESR-6300-NCP-K9

Slot Type State Insert time (ago)
--------- ------------------- --------------------- -----------------
0 ESR-6300-NCP-K9 ok 07:30:08
0/0 ESR-6300-2x1GE ok 07:29:45
0/1 ESR-6300-ES-4 ok 07:29:45
R0 ESR-6300-NCP-K9 ok, active 07:31:20
F0 ESR-6300-NCP-K9 ok, active 07:31:20
P0 PWR-12V ok 07:30:46
```

show platform diag: Example

```
Router#show platform diag
Chassis type: ESR-6300-NCP-K9

Slot: 0, ESR-6300-NCP-K9
Running state : ok
Internal state : online
Internal operational state : ok
Physical insert detect time : 00:01:45 (07:30:11 ago)
Software declared up time : 00:01:16 (07:30:40 ago)
CPLD version :
Firmware version : 2.1(DEV) [vitu-vitu 127]

Sub-slot: 0/0, ESR-6300-2x1GE
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:02:07 (07:29:49 ago)
Logical insert detect time : 00:02:07 (07:29:49 ago)

Sub-slot: 0/1, ESR-6300-ES-4
Operational status : ok
Internal state : inserted
Physical insert detect time : 00:02:07 (07:29:49 ago)
Logical insert detect time : 00:02:07 (07:29:49 ago)

Slot: R0, ESR-6300-NCP-K9
Running state : ok, active
Internal state : online
Internal operational state : ok
Physical insert detect time : 00:00:33 (07:31:24 ago)
Software declared up time : 00:00:33 (07:31:24 ago)
CPLD version : 00000000
Firmware version : 2.1(DEV) [vitu-vitu 127]

Slot: F0, ESR-6300-NCP-K9
Running state : ok, active
Internal state : online
Internal operational state : ok
Physical insert detect time : 00:00:33 (07:31:24 ago)
Software declared up time : 00:01:42 (07:30:14 ago)
Hardware ready signal time : 00:00:00 (never ago)
Packet ready signal time : 00:01:45 (07:30:11 ago)
CPLD version : 00000000
Firmware version : 2.1(DEV) [vitu-vitu 127]

Slot: P0, PWR-12V
State : ok
Physical insert detect time : 00:01:06 (07:30:50 ago)

Slot: GE-POE, Unknown
State : NA
Physical insert detect time : 00:00:00 (never ago)
```

show platform software status control-processor: Example

```
Router# show platform software status control-processor
RP0: online, statistics updated 9 seconds ago
Load Average: healthy
1-Min: 0.32, status: healthy, under 5.00
5-Min: 0.33, status: healthy, under 5.00
```

```
15-Min: 0.35, status: healthy, under 5.00
Memory (kb): healthy
Total: 3959840
Used: 2894588 (73%), status: healthy
Free: 1065252 (27%)
Committed: 2435656 (62%), under 90%
Per-core Statistics
CPU0: CPU Utilization (percentage of time spent)
User: 0.50, System: 0.91, Nice: 0.00, Idle: 98.07
IRQ: 0.40, SIRQ: 0.10, IOwait: 0.00
CPU1: CPU Utilization (percentage of time spent)
User: 0.81, System: 0.30, Nice: 0.00, Idle: 98.48
IRQ: 0.20, SIRQ: 0.20, IOwait: 0.00
CPU2: CPU Utilization (percentage of time spent)
User: 0.81, System: 2.65, Nice: 0.00, Idle: 95.41
IRQ: 1.12, SIRQ: 0.00, IOwait: 0.00
CPU3: CPU Utilization (percentage of time spent)
User: 7.66, System: 17.05, Nice: 0.00, Idle: 70.58
IRQ: 4.59, SIRQ: 0.10, IOwait: 0.00
Router#
```

show diag slot R0 eeprom detail: Example

```
Router#show diag slot R0 eeprom detail
Slot R0 EEPROM data:

EEPROM version : 4
Compatible Type : 0xFF
Controller Type : 3497
Hardware Revision : 0.2
PCB Part Number : 73-101841-01
Board Revision : 07
Deviation Number : 0
Fab Version : 01
PCB Serial Number : FOC23032UUN
Top Assy. Part Number : 068-102311-01
Top Assy. Revision : 01
Chassis Serial Number : FOC23032UUN
Deviation Number : 0
RMA Test History : 00
RMA Number : 0-0-0-0
RMA History : 00
Product Identifier (PID) : ESR-6300-NCP-K9
Version Identifier (VID) : V00
CLEI Code : UNASSIGNED
Manufacturing Test Data : 00 00 00 00 00 00 00 00
Field Diagnostics Data : 00 00 00 00 00 00 00 00
Chassis MAC Address : d478.9b5c.d680
MAC Address block size : 128
Asset ID :
Asset Alias :
Unknown Field (type 0086): 00 00 00 00
Asset ID :
```

show version: Example

```
Router# show version
Cisco IOS XE Software, Version 17.01.01
Cisco IOS Software [Amsterdam], ISR Software (ARMV8EL_LINUX_IOSD-UNIVERSALK9_IOT-M), Version
 17.1.1, RELEASE SOFTWARE (fc1)
```

```
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2019 by Cisco Systems, Inc.
Compiled Tue 19-Nov-19 06:19 by mcpre


Cisco IOS-XE software, Copyright (c) 2005-2019 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.


ROM: IOS-XE ROMMON

Router uptime is 1 day, 49 minutes
Uptime for this control processor is 1 day, 51 minutes
System returned to ROM by reload
System image file is "bootflash:c6300-universalk9.17.01.01.SPA.bin"
Last reload reason: Reload Command



This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.


Technology Package License Information:

------------------------------------------------------------------
Technology    Technology-package           Technology-package
              Current       Type           Next reboot
------------------------------------------------------------------
network-stack    network-advantage   Smart License    network-advantage
dna-stack        dna-advantage       Smart License    dna-advantage

The current throughput level is 250000 kbps


Smart Licensing Status: UNREGISTERED/EVAL MODE

cisco ESR-6300-CON-K9 (1RU) processor with 705559K/6147K bytes of memory.
Processor board ID FOC233501BF
MCU bootloader version: 8
MCU application version: 10
1 Virtual Ethernet interface
6 Gigabit Ethernet interfaces
32768K bytes of non-volatile configuration memory.
4194304K bytes of physical memory.
```

```
60542080K bytes of USB Flash at usbflash0:.
6500351K bytes of Bootflash at bootflash:.

Configuration register is 0x2102
```

show power: Example

```
Router# show power
Main PSU :
Total Power Consumption from 3.3V Line : 0.34
Total Power Consumption from 5V Line : 6.64
Configured Mode : N/A
Current runtime state same : N/A
PowerSupplySource : External PS
POE Module :
Configured Mode : N/A
Current runtime state same : N/A
Total power available : 120 Watts
```

SNMP Polling of Temperature OID

Support has been added for SNMP MIB to be able to return values from temperature sensors. The output should look similar to the **show environment** CLI.

The output of a **show environment**:

```
router#show environment

Number of Critical alarms:  0
Number of Major alarms:     0
Number of Minor alarms:     0

 Slot        Sensor          Current State   Reading
Threshold(Minor,Major,Critical,Shutdown)
 ----------  --------------  --------------- ------------
----------------------------------------
 R0          Temp: TS1       Normal          42    Celsius      (75 ,80 ,90 ,na )(Celsius)
 R0          Temp: TS2       Normal          37    Celsius      (75 ,80 ,90 ,na )(Celsius)
```

The output from an snmpwalk would look similar to this:

```
[root@sg-centos-hv ~]# snmpwalk -v 2c -c public 33.33.33.204 1.3.6.1.4.1.9.9.13.1.3.1
SNMPv2-SMI::enterprises.9.9.13.1.3.1.2.1 = STRING: "Sensor 1"
SNMPv2-SMI::enterprises.9.9.13.1.3.1.3.1 = Gauge32: 48
SNMPv2-SMI::enterprises.9.9.13.1.3.1.4.1 = INTEGER: 93
SNMPv2-SMI::enterprises.9.9.13.1.3.1.5.1 = INTEGER: 0
SNMPv2-SMI::enterprises.9.9.13.1.3.1.6.1 = INTEGER: 1
SNMPv2-SMI::enterprises.9.9.13.1.3.1.7.1 = INTEGER: 0
```

The ciscoEnvMonTemperatureStatusEntry oid is 1.3.6.1.4.1.9.9.13.1.3.1:

- ciscoEnvMonTemperatureStatusIndex (.1)
- ciscoEnvMonTemperatureStatusDescr (.2)
- ciscoEnvMonTemperatureStatusValue (.3)
- ciscoEnvMonTemperatureThreshold (.4)

- ciscoEnvMonTemperatureLastShutdown (.5)
- ciscoEnvMonTemperatureStatus (.6)

Technical Assistance

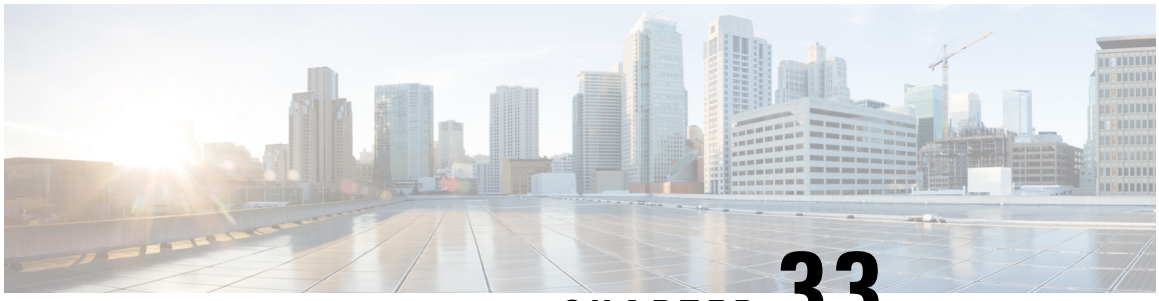
Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services,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

Additional References

The following sections provide references related to the power efficiency management feature.

MIBs

MIBs	MIBs Link
CISCO-ENTITY-FRU-CONTROL-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use the Cisco MIB Locator at: http://www.cisco.com/go/mibs.



CHAPTER 33

Additional Information

This chapter contains the following sections:

- Where to find Additional Information, on page 345
- Communications, Services, and Additional Information, on page 345

Where to find Additional Information

The ESR6300 offers a rich IOS-XE feature set. This marketing data sheet provides a complete list of all of the features. https://www.cisco.com/c/dam/en/us/products/routers/cisco-embedded-series-router-c78-742901.pdf

Previous chapters in this guide provided an introduction the the ESR6300, as well as some of the basic configuration and feature differences for this product. The IOS-XE Operating System runs on numerous routing devices, and as such, has a wealth of additional configuration information.

Communications, Services, and Additional Information

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

Cisco Bug Search Tool

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

Documentation Feedback

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