Cisco Cable Modem High-Speed WAN Interface Cards Configuration Guide

This document describes how to configure Cisco Data-Over-Cable Service Interface Specification (DOCSIS) cable modem high-speed WAN interface cards (HWICs) in the following supported Cisco routers: Cisco IAD2431 integrated access devices; Cisco 2691, Cisco 3725, Cisco 3745 series routers; Cisco 815, Cisco 1800, Cisco 2800, and Cisco 3800 integrated services routers (ISRs).

Cisco cable modem HWICs are designed to be fully compliant with DOCSIS 2.0 standards in the United States, Europe, and Japan. Cisco cable modem HWICs provide secure, high-speed connections to hybrid fiber-coaxial (HFC) cable networks.

The Cisco cable modem HWICs allow the router to communicate over high-speed data (HSD) cable networks for office-to-Internet connectivity or for branch-to-branch connectivity. Supported on a wide range of platforms, the Cisco cable modem HWICs are suitable for installations ranging from small office/home office (SOHO) to small and medium business (SMB) to enterprise branch offices. When the Cisco cable modem HWIC is combined with the powerful Cisco IOS software and Cisco's wide range of industry-leading access routers, an unparalleled range of services possible, all within a single, easily manageable platform. This combination allows a provider or business to minimize operational expenses while maximizing the potential return on invested capital.

Note The Cisco cable modem HWIC is fully DOCSIS 2.0 compliant. To see the DOCSIS 2.0 U.S. requirements and specifications, see the CableLabs website at http://www.cablemodem.com/specifications/specifications20.html

To see Euro DOCSIS 2.0 requirements, see the ComLabs website at http://www.tcomlabs.com

Feature History for Cisco Cable Modem HWICs (HWIC-CABLE-D-2, HWIC-CABLE-E/J-2)

Release	Modification
12.4(11)T (router software)	This feature was introduced.
12.4(6)XE (router software)	This command was integrated into Cisco IOS Release 12.4(6)XE.

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Open Source License Acknowledgements

The following notices pertain to this software license.

OpenSSL/Open SSL Project

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

License Issues

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License:

Copyright © 1998-2007 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment: “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”.
4. The names “OpenSSL Toolkit” and “OpenSSL Project” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called “OpenSSL” nor may “OpenSSL” appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:

   “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”.

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT “AS IS”' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License:

Copyright © 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).

The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young’s, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   “This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)”.

The word ‘cryptographic’ can be left out if the routines from the library being used are not cryptography-related.

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: “This product includes software written by Tim Hudson (tjh@cryptsoft.com)”.

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License].

Restrictions for the Cisco Cable Modem HWICs

The Cisco IOS software version and feature set software that are installed on the host router must be compatible with the cable modem HWIC. See the “Feature History for Cisco Cable Modem HWICs (HWIC-CABLE-D-2, HWIC-CABLE-E/J-2)” section on page 1. To view the Cisco IOS software release and router feature set, enter the **show version** command in privileged EXEC mode.

Note To configure the Cisco IOS software on your router, see the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4* at the following URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_book09186a0080430ee6.html

Information About the Cisco Cable Modem HWICs

This section describes the features of and some important concepts about Cisco cable modem HWICs:

Accessibility

These HWICs can be configured using the Cisco command-line interface (CLI). The CLI conforms to accessibility code 508 because it is text based and because it relies on a keyboard for navigation. All functions of the router can be configured and monitored through the CLI.

For a complete list of guidelines and Cisco products adherence to accessibility, see Cisco Accessibility Products at the following URL:

http://www.cisco.com/web/about/responsibility/accessibility/products

Hardware Overview

The two types of Cisco cable modem HWICs are as follows:

- HWIC-CABLE-D-2

  HWIC-CABLE-D-2 is the cable modem HWIC that is designed for North American customers.

- HWIC-CABLE-E/J-2

  HWIC-CABLE-E/J-2 is the cable modem HWIC that is designed for European and Japanese customers.

Note For complete information about Cisco cable modem HWIC hardware, see the *Cisco Network Modules Hardware Installation Guide at the following URL:*

http://www.cisco.com/en/US/products/hw/modules/ps2797/products_module_installation_guide_book09186a0080692a92.html

Platform Support for Cisco Cable Modem HWICs

Cisco cable modem HWICs can be inserted into WIC or HWIC slots. Table 1 lists the Cisco routers that support WICs and HWICs.

Note A maximum of four Cisco cable modem HWICs can be inserted in the chassis, depending on the availability of chassis slots.

Table 1 Cisco Router Support for WICs and HWICs

Cisco Router	WIC	HWIC
815[1] **Note** The HWIC-CABLE-D-2 is a fixed-configuration card and is not field replaceable. The HWIC-CABLE-D-2 card operates only in WIC mode with 8-Mbps throughput.	Yes	No
1800	No	Yes
IAD2431[1]	Yes	No
2691[1]	Yes	No
2800 series	No	Yes
3700 series[1]	Yes	No
3800 series	No	Yes

1. When the cable modem HWIC is placed in these routers, the HWIC operates only in WAN interface card (WIC) mode, providing total throughput of 8 Mbps on the cable modem HWIC.

Note For specific information about the routers that support the Cisco cable modem HWICs, see the hardware installation documentation for your router, which is available on http://www.cisco.com/.

Port Numbering Schemes

Table 2 shows the port number schemes used on the Cisco routers. For information about port numbering on interface cards in specific routers, see the *Cisco Interface Cards Installation Guide*.

Note For specific port numbering information for the routers that support the Cisco cable modem HWICs, see the hardware installation documentation for your router, which is available on http://www.cisco.com/.

Table 2 Port Numbering on the Cisco Routers

Cisco Router	Interface Numbering
1841, 2800, and 3800 ISRs	x/y/z
IAD2431, 2691, 3725, 3745, and 1800 ISR	x/y
815 ISR	x

Note The slot number for all WIC interfaces on Cisco ISRs is always 0. (The W0 and W1 slot designations are for physical slot identification only.) Interfaces in the WICs are numbered from right to left, starting with 0/0 for each interface type, regardless of which physical slot the WICs are installed in.

Note The slot for WICs on the Cisco 2430 IADs is numbered slot 0. WIC interfaces are numbered by interface with this slot number and an interface number, starting with 0 and continuing from right to left.

Software Features and Benefits

Cisco cable modem HWICs are configured automatically by the network (in compliance with DOCSIS provisioning specifications). The configuration file is defined and generated by the cable service provider and delivered over the WAN/DOCSIS network through the radio frequency (RF) interface on the Cisco cable modem HWIC installed in the router. The HWIC provides a path from the router to the service provider network-based DHCP server for host address assignment on the Cisco cable modem HWIC and on the WAN interface of the router.

Note Cisco cable modem HWICs are fully DOCSIS 2.0 compliant. For DOCSIS 2.0 requirements, see the CableLabs website, whick is available at the following URL:

http://www.cablemodem.com/specifications/specifications20.html

The Cisco cable modem HWICs provide the following features and benefits.

Note The following benefits assume that a full-featured enterprise router is in use, rather than use of a Cisco cable modem HWIC as a bridge.

- Provides quality of service (QoS) upstream flow control, integrating DOCSIS QoS with Cisco IOS software QoS and packet cable multimedia (PCMM) architecture QoS with Cisco IOS software QoS
- Leverages Cisco IOS software to deliver advanced network services and applications
- Supports compression and decompression algorithms (codecs)

How to Configure the Router to Interact with the Cable Modem

This section describes how to configure the host router when interacting with the Cisco cable modem HWIC:

- Configuring Bridging, page 8
- Configuring Routing, page 10
- Configuring Network Address Translation, page 11
- Configuring Dynamic Host Configuration Protocol, page 11
- Configuring QoS, page 12
- Configuring Easy Virtual Private Network, page 15
- Configuring Multicast with IGMP Proxy, page 15
- Configuring Circuit Emulation over IP, page 21

Cisco cable modem HWICs are configured automatically through a configuration file that is generated by the cable service provider. You can configure the router to function either as a bridge or as a router. The following sections briefly describe both applications.

Note To configure Cisco IOS software on your router, see the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4*, which is available at the following URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_book09186a0080430ee6.html

Note The Cisco cable modem HWICs are fully DOCSIS 2.0 compliant. To see DOCSIS 2.0 requirements, see the CableLabs website, which is available at the following URL:

http://www.cablemodem.com/specifications/specifications20.html

Configuring Bridging

Cisco cable modem HWICs comply with the Multimedia Cable Network System Partners Ltd. Consortium (MCNS) standard for interoperable cable modems; it supports full transparent bridging as well as DOCSIS-compliant transparent bridging.

To configure bridging between the router and the cable modem, perform the following tasks, beginning in global configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge irb**
4. **bridge** *bridge-group* **protocol**
5. **bridge** *bridge-group* **route** *protocol*
6. **interface bvi** *bridge-group*
7. **interface** *port-type port-number*
8. **no ip address**
9. **bridge-group** *bridge-group*
10. **interface** *port-type port-number*
11. **no ip address dhcp client-id** *interface-name* **hostname** *host-name*
12. **bridge-group** *bridge-group*
13. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router# enable`
	`Router#`
		Enters privileged EXEC mode.

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

Step 3	`bridge irb`
	
	Example:
	`Router(config)# bridge irb`
	`Router(config)#`
		Enables Cisco IOS software to route a given protocol between routed interfaces and bridge groups or to route a given protocol between bridge groups.

	Command or Action	Purpose

Step 4	**bridge** *bridge-group* **protocol**
	
	Example:
	`Router(config)# bridge 59 protocol ieee`
	`Router(config)#`
		Defines the type of Spanning Tree Protocol.

Step 5	**bridge** *bridge-group* **route** *protocol*
	
	Example:
	`Router(config)# bridge 59 route ip`
	`Router(config)#`
		Enables the routing of a specified protocol in a specified bridge group.

Step 6	**interface bvi** *bridge-group*
	
	Example:
	`Router(config)# interface bvi 59`
	`Router(config-if)#`
		Creates the bridge-group virtual interface (BVI) that represents the specified bridge group to the routed world and links the corresponding bridge group to the other routed interfaces.

Step 7	**interface** *port-type port-number*
	
	Example:
	`Router(config-if)# interface gigabit ethernet 0/1`
	`Router(config-if)#`
		Enters interface configuration mode for the Ethernet 0 interface.

Step 8	**no ip address**
	
	Example:
	`Router(config-if)# no ip address`
	`Router(config-if)#`
		Disables the IP address of the coaxial cable interface, if an address has been set. IP address assignment happens if **ip address dhcp** is set and the IP address is not assigned by the second router. The address comes from the DHCP server.
		
		Note An IP address is not normally needed becaise bridging is a Layer 2 operation, so IP address is not normally needed.

Step 9	**bridge-group** *bridge-group*
	
	Example:
	`Router(config-if)# bridge-group 59`
	`Router(config-if)#`
		Assigns the Ethernet 0 interface to a bridge group. The bridge group must be an integer between 1 and 63.

Step 10	**interface** *port-type port-number*
	
	Example:
	`Router(config)# interface cable 0/2/0`
	`Router(config-if)#`
		Enters interface configuration mode for the Ethernet 0 interface.

Step 11	**no ip address dhcp**
	
	Example:
	`Router(config-if)# no ip address`
	`Router(config-if)#`
		Sets the **no** form of the **ip address dhcp** command to acquire an IP address on an interface from the DHCP.

	Command or Action	Purpose

Step 12	**bridge-group** *bridge-group*
	
	Example:
	`Router(config-if)# bridge-group 59`
	`Router(config-if)#`
		Assigns the Ethernet 0 interface to a bridge group. The bridge group must be an integer between 1 and 63.

Step 13	**end**
	
	Example:
	`Router(config-if)# end`
	`Router(config)#`
		Returns to global configuration mode.

Configuring Routing

Routing for the Cisco cable modem HWIC is on by default. To bring the Cisco cable modem HWIC online, use the **interface Cable-Modem** and **ip address dhcp** commands.

Note To bring the Cisco cable modem HWIC online, the cable modem must be in the *no shut down* state.

To configure routing between the router and the cable modem, perform the following tasks, beginning in global configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface cable-modem** ***number***
4. **ip address dhcp** *interface-name* **hostname** *host-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router# enable`
	`Router#`
		Enters privileged EXEC mode.

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

	Command or Action	Purpose

Step 3	**interface cable-modem** *number*
	
	Example:
	`Router(config)# interface Cable-Modem 0`
	`Router(config-if)#`
		Enters interface configuration mode for the cable modem interface.

Step 4	**ip address dhcp** *interface-name* hostname *host-name*
	
	Example:
	`Router(config-if)# ip address dhcp`
	`Router(config-if)#`
		Acquires an IP address and allows any interface to dynamically learn its IP address by using the DHCP protocol.

Step 5	**end**
	
	Example:
	`Router(config-if)# end`
	`Router(config)#`
		Exits interface configuration mode for the cable modem interface.

Configuring Network Address Translation

Network Address Translation (NAT) operates on a router that is connecting two networks; one of these networks (designated as the *inside network*) is addressed with either private or obsolete addresses that must be converted into legal addresses before it forwards packets to the other network (designated as the *outside network*). The translation operates in conjunction with routing, so that NAT can simply be enabled on a customer-side Internet access router when translation is desired.

Note To configure NAT on your router, see the NAT documentation, which is available at the following URL:

http://www.cisco.com/en/US/tech/tk648/tk361/tk438/tsd_technology_support_sub-protocol_home.html

Configuring Dynamic Host Configuration Protocol

As explained in RFC 2131, Dynamic Host Configuration Protocol (DHCP) provides configuration parameters to Internet hosts. DHCP consists of two components: a protocol for delivering host-specific configuration parameters from a DHCP server to a host, and a mechanism for allocating network addresses to hosts. DHCP is built on a client/server model, in which designated DHCP server hosts allocate network addresses and deliver configuration parameters to dynamically configured hosts. By default, Cisco routers that are running Cisco IOS software simultaneously run DHCP server and relay agent software.

Note To configure DHCP on your router, see the *Configuring DHCP* documentation, which is available at the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca75c.html

Configuring QoS

Cisco cable modem HWICs have the ability to transmit congestion notification for the primary flow, as defined by the configuration received from the cable modem termination system (CMTS). The primary flow is for traffic that has the lowest priority. With this notification, Cisco IOS software performs QoS to manage congestion for primary flow traffic.

The remaining traffic going to secondary service flows is handed directly to the Cisco cable modem HWIC. During this process, the traffic bypasses the Cisco IOS software QoS classification or queuing mechanisms established by the Cisco cable modem HWIC. The Cisco cable modem HWIC then relays the CMTS policies to Cisco IOS software. Cisco IOS software then parses the classification parameters and defines an ACL that will match any non-primary flow traffic. This ACL is invoked before the Cisco IOS QoS classification step in the Cisco Express Forwarding (CEF) egress feature path.

With this functionality, class maps can be defined by using parameters that subclassify the primary flow traffic.

Note When congestion occurs on the primary flow, QoS queues traffic based on this class map.

To configure QoS between the router and the cable modem, perform the following tasks, beginning in global configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef**
4. **class-map** *match-all match-any class-map-name*
5. **match dscp** *dscp-value*
6. Repeat Step 2 and Step 3 for as many class maps and DSCP values as necessary.
7. **policy-map** *policy-map-name*
8. **class** *class-name* **class-default**
9. **bandwidth percent** *percentage*
10. Repeat Step 7 for as many classes as necessary.
11. **interface cable-modem** *number*
12. **service-flow primary** *upstream*
13. **service-policy output** *policy-map-name*

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router# enable`
	`Router#`
		Enters privileged EXEC mode.

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

Step 3	**ip cef**
	
	Example:
	`Router(config)# ip cef`
	`Router(config)#`
		Enables Cisco Express Forwarding (CEF) on the route processor card. Use the **ip cef** command in global configuration mode.

Step 4	**class map [match-all | match-any]** *class-map-name*
	
	Example:
	`Router(config)# class-map match-any VOICE`
	`Router(config-cmap)#`
		Specifies the name of the class for which you want to create or modify class map match criteria.

Step 5	**match dscp**
	
	Example:
	`Router(config-cmap)# match ip dscp ef`
	`Router(config)#`
		Identifies a specific IP Differentiated Services Code Point (DSCP) value as a match criterion.
		
		Note This command replaces the **match ip dscp** command.

Step 6	Repeat Step 2 and Step 3 for as many class maps and DSCP values as necessary.	

Step 7	**policy-map**
	
	Example:
	`Router(config)# policy-map V3PN-teleworker`
	`Router(config-pmap)#`
		Specifies the name of the policy map to be created, added to, or modified before you can configure policies for classes whose match criteria are defined in a class map.

Step 8	**class** *class-name* **class-default**
	
	Example:
	`Router(config-pmap)# class CALL-SETUP`
	`Router(config-pmap-c)#`
		Specifies the name of the class whose policy you want to create or change or to specify the default class (commonly known as the *class-default class*).

Step 9	**bandwidth percent** *percentage*
	
	Example:
	`Router(config-pmap)# bandwidth percent 2`
	`Router(config-pmap-c)#`
		Specifies or modifies the bandwidth allocated for a class that belongs to a policy map.

Step 10	Repeat Step 7 for as many classes as necessary.	

	Command or Action	Purpose

Step 11	**interface Cable-Modem** *port-number*
	
	Example:
	`Router(config)# interface Cable-Modem 0/0/1`
	`Router(config-if)#`
		Specifies the port to attach to the policy map, and enters interface configuration mode. Valid interfaces include physical ports.

Step 12	**service-flow primary** *upstream*
	
	Example:
	`Router(config-if)# service-flow primary upstream`
	`Router(config-if)#`
		Specifies whether the primary service flow is set to upstream traffic. Only secondary service flows can be configured.

Step 13	**service-policy output** *policy-map-name*
	
	Example:
	`Router(config-if)# service-policy output anyname`
	`Router(config)#`
		Attaches a policy map to the output interface or virtual circuit (VC), to be used as the service policy for that interface or VC.

Examples

The following example shows configuration of QoS on the router.

Identify the class to which you want to apply QoS. In this example, the voice class is identified by the alphanumeric characters **ef**:

```
Router(config)# ip cef
class-map match-all VOICE
match ip dscp ef
class-map match-any CALL-SETUP
match ip dscp af31
match ip dscp cs3
class-map match-any INTERNETWORK-CONTROL
match ip dscp cs6
```

The following example specifies the priority assigned to the different classes. Voice is assigned the highest priority in this example:

```
Router(config)# policy-map anyname
class CALL-SETUP
bandwidth percent 2
class INTERNETWORK-CONTROL
bandwidth percent 5
class VOICE
priority 234
class class-default
fair-queue
random-detect
interface Cable-Modem0/2/0
ip address dhcp
service-module ip address 209.165.200.225 255.255.255.224
```

Use the **interface Cable-Modem** command to apply the priority policy to the cable modem interface:

```
Router(config)# interface Cable-Modem0/2/0
service-flow primary up
service-policy output anyname
```

Use the **show ip access-lists dynamic** command to view the dynamic IP access lists:

```
Router# show ip access-lists dynamic
Extended IP access list CM_SF#1
10 permit udp any any eq 5060 (650 matches)
20 permit tcp any any eq 5060
30 permit udp any any dscp ef (806184 matches)
c2801-61#
```

Configuring Easy Virtual Private Network

VPN provides security by performing a high level of authentication and by encrypting the data between two particular endpoint routers. Establishing a VPN connection between two routers can be complicated; it typically requires tedious coordination between network administrators to configure the VPN parameters of the two routers.

The Cisco Easy VPN remote feature eliminates much of this tedious work by implementing Cisco Unity Client Protocol, which allows most VPN parameters to be defined at a Cisco IOS Easy VPN server.

After the Easy VPN server has been configured, a VPN connection can be created with minimal configuration on an Easy VPN remote router. When the Easy VPN remote router initiates the VPN tunnel connection, the Cisco Easy VPN server pushes the IPsec policies to the Easy VPN remote and creates the VPN tunnel connection.

To learn more about configuring Easy VPN, see *Configuration Example: Easy VPN*, which is available at the following URL:

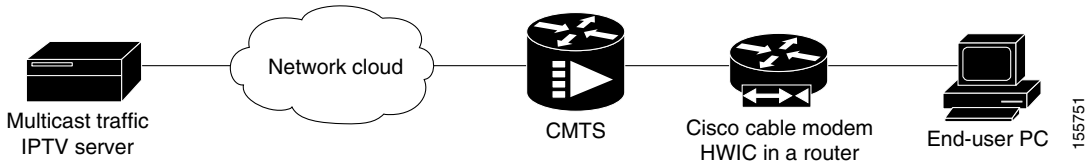
http://www.cisco.com/en/US/products/ps5854/prod_configuration_guide09186a00802c3270.html

Configuring Multicast with IGMP Proxy

The Internet Group Management Protocol (IGMP) proxy mechanism permits hosts that are not directly connected to a downstream router to join a multicast group sourced from an upstream network.

Figure 1 shows a typical multicast configuration.

Figure 1 Typical Multicast Configuration



Note For additional information about configuring IGMP proxy, see the IGMP proxy configuration document, which is available at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtudlr.htm#1020541

Prerequisites

The Cisco cable modem HWIC can be configured for multicast with IGMP proxy.

Using a DOCSIS cable modem configurator tool, specify the following fields in the ASCII configuration file:

```
42          = 01 00 5e 00 00 09
42          = 01 00 5e 00 00 0d
42          = 01 00 5e 00 01 27
42          = 01 00 5e 00 01 28


=====================================================================
  CM MIC        = b5 22 c0 24 5d 8e 64 97 93 e0 94 35 f8 a6 3e 53
CMTS MIC        = 72 c0 d2 d8 01 67 d5 57 5b 7c 91 df 00 6d 9e 71
=====================================================================
```

Note For a complete list of common radio frequency interface encodings, see the *DOCSIS 2.0 Radio Frequency Interface Specification* document, which is available on the CableLabs website at the following URL:

http://www.cablemodem.com/downloads/specs/CM-SP-RFI2.0-I10-051209.pdf

To configure multicast with IGMP proxy, perform the following tasks.

SUMMARY STEPS

1. **enable**
2. **show ip mroute**
3. **show interfaces** *type number*
4. **show ip igmp membership** *group-address group-name*
5. **show ip pim** *vrf vrf-name neighbor interface-type interface-number*
6. **show running-config** *options*
7. **configure terminal**
8. **ip multicast-routing distributed**
9. **ip igmp helper-address** *ip address*
10. **ip igmp proxy-service** *ip address*
11. **ip pim** *sparse-dense-mode*
12. **ip igmp mroute-proxy** *type number*
13. **ip pim rp-address** *rp-address access-list*

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router# enable`
	`Router#`
		Enters privileged EXEC mode.

Step 1	**`show ip mroute`**
	
	Example:
	`Router# show ip mroute`
	`Router#`
		Displays the contents of the IP multicast routing table.

Step 2	**`show interfaces`** *`type number`*
	
	Example:
	`Router# show interfaces c0`
	`Router#`
		Displays statistics for all interfaces configured on the router.

Step 3	**`show ip igmp membership`** *`group-address group-name`*
	
	Example:
	`Router> show ip igmp membership`
	`Router>`
		Displays Internet Group Management Protocol (IGMP) membership information for multicast groups and (S, G [channel or multicast group filtering entry]) channels.

Step 4	**`show ip pim`** *`vrf vrf-name`* `neighbor` *`interface-type interface-number`*
	
	Example:
	`Router# show ip pim neighbor`
	`Router#`
		Lists the Protocol Independent Multicast (PIM) neighbors discovered by the Cisco IOS software.

Step 5	**`show running-config`** *`options`*
	
	Example:
	`Router# show running-config`
	`Router#`
		Displays the contents of the currently running configuration file or the configuration for a specific class map, interface, map class, policy map, or virtual circuit (VC) class.

Step 6	**`configure terminal`**
	
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 7	**`ip multicast-routing`** *`distributed`*
	
	Example:
	`Router(config)# ip multicast-routing`
	`Router(config)#`
		Enables IP multicast routing.

	Command or Action	Purpose

Step 8	**ip igmp helper-address** *ip address*
	
	Example:
	`Router(config-if)# ip igmp helper-address 209.165.201.1`
	`Router(config-if)#`
		Causes the system to forward all Internet Group Management Protocol (IGMP) host reports and leave messages received on the interface to the specified IP address.

Step 9	**ip igmp proxy-service** *ip address*
	
	Example:
	`Router(config-if)# ip igmp proxy-service`
	`Router(config-if)#`
		Enables the mroute proxy service. Based on the IGMP query interval, the router periodically checks the mroute table for forwarding entries (*, G) that match interfaces configured with the **ip igmp mroute-proxy** command. Where there is a match, an IGMP report is created and received on this interface.

Step 10	**ip pim** ***s****parse-dense-mode*
	
	Example:
	`Router(config-if)# ip pim sparse-dense-mode`
	`Router(config-if)#`
		Treats the interface in either sparse mode or dense mode of operation, depending on which mode the multicast group operates in.

Step 11	**ip igmp mroute-proxy** *type number*
	
	Example:
	`Router(config-if)# ip igmp mroute-proxy Loopback0`
	`Router(config-if)#`
		Enables IGMP report forwarding of proxied (*, G) mroute entries.

Step 12	**ip pim rp-address** *rp-address access-list*
	
	Example:
	`Router(config)# ip pim rp-address 209.165.202.130`
	`Router(config)#`
		Specifies the IP address of a router to be a PIM RP address. This is a unicast IP address in four-part dotted-decimal notation.

Examples

The following example shows configuration of the router with multicast and IGMP proxy.

```
Router# show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 02:14:42/stopped, RP 209.165.202.130, flags: SJC
  Incoming interface: Cable-Modem0, RPF nbr 209.165.201.1
  Outgoing interface list:
    Vlan1, Forward/Sparse-Dense, 02:14:42/00:02:51

(209.165.200.226, 224.1.1.1), 02:14:21/00:02:50, flags: JT
  Incoming interface: Cable-Modem0, RPF nbr 209.165.201.1
  Outgoing interface list:
```

```
    Vlan1, Forward/Sparse-Dense, 02:14:21/00:02:51

(*, 224.0.1.40), 21:03:48/00:02:40, RP 209.165.202.130, flags: SJCL
  Incoming interface: Cable-Modem0, RPF nbr 209.165.201.1
  Outgoing interface list:
    Loopback0, Forward/Sparse-Dense, 21:03:48/00:02:40

Router# show interfaces c0
Cable-Modem0 is up, line protocol is up
  HFC state is OPERATIONAL, HFC MAC address is 00d0.2bfe.66ea
  Hardware is Cable modem, address is 0014.a875.8dec (bia 0014.a875.8dec)
  Internet address is 209.165.201.130
  MTU 1500 bytes, BW 2000 Kbit, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 21/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: Class-based queueing
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 1/1 (allocated/max allocated)
     Available Bandwidth 520 kilobits/sec
  30 second input rate 2961000 bits/sec, 243 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
  HFC input: 0 errors, 0 discards, 0 unknown protocols 0 flow control discards
  HFC output: 0 errors, 0 discards
     11299559 packets input, 4245935967 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     9 input errors, 0 CRC, 0 frame, 9 overrun, 0 ignored
     0 input packets with dribble condition detected
     59044 packets output, 6089309 bytes, 0 underruns
     0 output errors, 0 collisions, 32 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out

Router# show ip igmp membership
Flags: A  - aggregate, T - tracked
       L  - Local, S - static, V - virtual, R - Reported through v3
       I - v3lite, U - Urd, M - SSM (S,G) channel
       1,2,3 - The version of IGMP the group is in
Channel/Group-Flags:
       / - Filtering entry (Exclude mode (S,G), Include mode (*,G))
Reporter:
       <mac-or-ip-address> - last reporter if group is not explicitly tracked
       <n>/<m>      - <n> reporter in include mode, <m> reporter in exclude

 Channel/Group                  Reporter        Uptime   Exp.  Flags  Interface
 *,224.1.1.1                    172.16.0.33 02:14:51 02:09 2A     Lo0
*,224.0.1.40                   172.16.0.33 21:04:16 02:12 2LA    Lo0

Router# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.0.0.1          Cable-Modem0             19:49:29/00:01:29 v2    16384/ DR S

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

```
Current configuration : 4362 bytes
!
! Last configuration change at 23:48:55 PST Mon Feb 27 2006
! NVRAM config last updated at 23:48:56 PST Mon Feb 27 2006
!
version 12.4
service timestamps debug datetime localtime
service timestamps log datetime localtime
no service password-encryption
service internal
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging buffered 500000 debugging
no logging console
enable password lab
!
no aaa new-model
!
resource policy
!
no ip dhcp use vrf connected
!
no ip domain lookup
ip multicast-routing
!
interface Loopback0
 ip address 172.16.0.33 255.255.255.255
 ip pim sparse-dense-mode

ip igmp helper-address 209.165.201.1
 ip igmp proxy-service
!
interface FastEthernet0
 ip address 172.16.5.203 255.255.255.0
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet1
 load-interval 30
 duplex full
 speed 100
!
interface FastEthernet2
 load-interval 30
!
interface FastEthernet4
 load-interval 30
!
interface Cable-Modem0
 ip address dhcp
ip pim sparse-dense-mode
 load-interval 30
 no keepalive
!
interface Vlan1
 ip address 192.168.129.1 255.255.255.0
 ip pim sparse-dense-mode
 ip igmp mroute-proxy Loopback0
 load-interval 30
```

```
!
router rip
 version 2
 network 209.165.201.0
 network 192.168.129.0
 no auto-summary
!
ip route 192.168.101.0 255.255.255.0 10.0.0.200
ip route 172.16.6.254 255.255.255.255 192.168.1.1
!
no ip http server
no ip http secure-server
ip pim rp-address 209.165.202.130
!
end
```

Configuring Circuit Emulation over IP

Circuit Emulation over IP (CEoIP) provides protocol-independent transport over IP networks. It enables proprietary or legacy applications to be carried transparently to the destination, in a manner similar to that of a leased line.

Note For the multiple service operator (MSO), Cisco recommends using a uBR7246VXR with MC28U line card to configure CEoIP on the Cisco cable modem HWIC.

Note For more information about CEoIP configuration, see the *Circuit Emulation over IP* feature guide, which is available at the following URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a008045523e.html

Configuring the NM-CEM-4TE1 Card Type

Perform this task to configure the Cisco cable modem HWIC.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **card type** {**t1** | **e1**} *slot*
4. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

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

Step 3	**card type** {**t1** | **e1**} *slot*
	
	Example:
	`Router(config)# card type t1 1`
	`Router(config)#`
		Configures the card type by specifying the transmission mode for the ports on the network module.
		• All four ports on the CEoIP T1/E1 network module must operate in the same mode.
		• Use the **t1** or **e1** keyword to specify the transmission mode for all four ports.
		Note This command is entered only once, and changes do not take effect unless the **reload** command is used or the router is rebooted.

Step 4	**end**
	
	Example:
	`Router(config)# end`
	`Router#`
		Exits global configuration mode and returns to privileged EXEC mode.

Configuring the T1/E1 Line

Perform this task to configure the T1 or E1 line.

This task does not apply to the NM-CEM-4SER.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **controller** {**t1** | **e1**} *slot*/*port*
4. **framing** {**esf** | **sf** | **unframed**}
   or
   framing {**crc4** | **no-crc4** | **unframed**}
5. **clock source** {**internal** | **line** | **adaptive** *channel-number* [**closed-loop** | **open-loop** | **coarse**]}
6. **cablelength** {**long** *attenuation* | **short** *length*}
7. **crc-threshold** *value*
8. **description** *text*
9. **loopback**{**local** {**line** | **payload**} | **network**}
10. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

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

Step 3	**controller** {**t1** | **e1**} *slot*/*port*
	
	Example:
	`Router(config)# controller t1 1/0`
	`Router(config-controller)#`
		Enters controller configuration mode.
		• Use the *slot* and *port* arguments to specify the slot number and port number to be configured.

	Command or Action	Purpose

Step 4	**framing** {**esf** | **sf** | **unframed**}
	or
	framing {crc4 | no-crc4 | unframed}
	
	Example:
	`Router(config-controller)# framing esf`
	`Router(config-controller)#`
	
	Example:
	`Router(config-controller)# framing crc4`
	`Router(config-controller)#`
		(Optional) Configures the framing format for a T1 or E1 port to synchronize the port and the attached device.
		
		T1 Port Framing Options
		• Use the **esf** keyword to specify Extended Superframe as the T1 framing type.
		• Use the **sf** keyword to specify Superframe (also commonly called D4 framing) as the T1 framing type. This is the default.
		
		E1 Port Framing Options
		• Use the **crc4** keyword to specify the G.704 standard with the optional cyclic redundancy check 4 (CRC4) mechanism defined in time slot zero (0) enabled as the E1 framing type. This is the default.
		• Use the **no-crc4** keyword to specify the G.704 standard with the optional CRC4 mechanism defined in time slot zero (0) disabled as the E1 framing type.
		
		T1 or E1 Port Framing Option
		• Use the **unframed** keyword to specify the unchannelized mode of framing.
		
		Note If you do not configure framing, the framing on the customer premises equipment (CPE) devices on each end of the connection must match.

	Command or Action	Purpose

Step 5	**clock source** {**internal** | **line** | **adaptive** *channel-number* [**closed-loop** | **open-loop** | **coarse**]}
	
	Example:
	`Router(config-controller)# clock source adaptive 6`
	`Router(config-controller)#`
		Configures the clock source for a T1 or E1 port.
		• Use the **internal** keyword to specify that the port transmit clock (TxC) is derived from the time-division multiplexing (TDM) bus backplane clock, if one exists in the router, or from the onboard oscillator on the network module.
		• Use the **line** keyword to specify that the port transmit clock is derived from the receive clock (RxC) on the same port.
		• Use the **adaptive** keyword to specify that the port transmit clock is locally synthesized on the basis of the average data content of the dejitter buffer of one of the channels on this port. If the **adaptive** keyword is selected, use the *channel-number* argument to specify the channel whose dejitter buffer is to be used to synthesize the transmit clock of the port.
		– Use the **closed-loop** keyword to specify that the enhanced adaptive clock algorithm is used to improve the adaptive clock accuracy.
		– Use the **open-loop** keyword to specify that some of the enhancements to the adaptive clock algorithm are used to improve the adaptive clock accuracy.
		– Use the **coarse** keyword to specify that the original adaptive clock algorithm is used.
		Note The **closed-loop**, **open-loop**, and **coarse** keywords are supported only in Cisco IOS Release 12.4(2)T and later releases.

Step 6	**cablelength** {**long** *attenuation* | **short** *length*}
	
	Example:
	`Router(config-controller)# cablelength long -15db`
	`Router(config-controller)#`
		(Optional) Specifies the line build-out characteristics of the internal CSU on a T1 port.
		• Use the **long** keyword to specify that the signal characteristics are set for a long cable length. If the **long** keyword is selected, use the *attenuation* argument to specify the T1 signal attenuation.
		• Use the **short** keyword to specify that the signal characteristics are set for a short cable length. If the **short** keyword is selected, use the *length* argument to specify the T1 cable length.
		Note This command does not apply to an E1 port.

	Command or Action	Purpose

Step 7	**crc-threshold** *value*
	
	Example:
	`Router(config-controller)# crc-threshold 512`
	`Router(config-controller)#`
		(Optional) Configures the number of cyclical redundancy check (CRC) errors in one second that result in the second being declared a severely errored second (SES).
		• Use the *value* argument to specify the number of CRC errors. Range is from 0 to 3000. Default is 320.
		Note This command does not apply to an E1 port.

Step 8	**description** *text*
	
	Example:
	`Router(config-controller)# description T1 line to 3rd floor PBX`
	`Router(config-controller)#`
		(Optional) Specifies a text description of the port.

Step 9	**loopback** {**local** {**line** | **payload**}| **network**}
	
	Example:
	`Router(config-controller)# loopback network`
	`Router(config-controller)#`
		(Optional) Creates a loopback from a T1 or E1 port.
		• Use the **local** keyword to create a loopback for transmitting the information from a locally attached CPE back to the locally attached CPE.
		– If the **local** keyword is selected, use the **line** keyword to create a full physical-layer loopback of all bits, including data and framing.
		– If the **local** keyword is selected, use the **payload** keyword to create a loopback of the data in the individual time slots only. In this mode, framing bits are terminated on entry and regenerated on exit instead of being looped back. This mode is not available if the port is configured for **framing unframed**.
		• Use the **network** keyword to create a loopback for transmitting the data received over the network from a remotely attached CPE back to the remotely attached CPE.

Step 10	**end**
	
	Example:
	`Router(config-controller)# end`
	`Router#`
		Exits controller configuration mode and returns to privileged EXEC mode.

Creating Circuit Emulation Channels on the T1/E1 Line

Perform this task to create CEM channels on the T1 or E1 line.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **controller** {**t1** | **e1**} *slot*/*port*
4. **cem-group** *group-number* {**unframed** | **timeslots** *timeslot* [**speed** {**56** | **64**}]}
5. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

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

Step 3	**controller** {**t1**|**e1**} *slot*/*port*
	
	Example:
	`Router(config)# controller t1 1/0`
	`Router(config-controller)#`
		Enters controller configuration mode.
		• Use the *slot* and *port* arguments to specify the slot number and port number to be configured.

Step 4	**cem-group** *group-number* {**unframed** | **timeslots** *timeslot* [**speed** {**56** | **64**}]}
	
	Example:
	`Router(config-controller)# cem-group 6 timeslots 1-4,9,10 speed 64`
	`Router(config-controller)#`
		Creates a circuit emulation (CEM) channel from one or more time slots of a T1 or E1 line of an NM-CEM-4TE1.
		• The *group-number* keyword identifies the channel number to be used for this channel. For T1 ports, the range is 0 to 23. For E1 ports, the range is 0 to 30.
		• Use the **unframed** keyword to specify that a single CEM channel is being created, including all time slots and the framing structure of the line.
		• Use the **timeslots** keyword and the *timeslot* argument to specify the time slots to be included in the CEM channel. The list of time slots may include commas and hyphens with no spaces between the numbers.
		– Use the **speed** keyword to specify the speed of the channels by specifying the number of bits of each time slot to be used. This keyword applies only to T1 channels.

Step 5	**end**
	
	Example:
	`Router(config-controller)# end`
	`Router#`
		Exits controller configuration mode and returns to privileged EXEC mode.

Configuring the Connection Using the xconnect Command

Perform this task to configure a connection using the **xconnect** command.

This task applies to configuring CEoIP on both the NM-CEM-4TE1 and the NM-CEM-4SER.

Note To properly configure the CEoIP feature, two CEoIP network modules must use the same User Datagram Protocol (UDP) port number to communicate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cem** *slot/port/channel*
4. **xconnect** *remote-ip-address virtual-connect-ID* **encapsulation** *encapsulation-type*
5. **local ip address** *ip-address*
6. **local udp port** *port*
7. **remote udp port** *port*
8. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

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

Step 3	**cem** *slot*/*port*/*channel*
	
	Example:
	`Router(config)# cem 3/1/0`
	`Router(config-cem)#`
		Enters CEM configuration mode to configure CEM channels.
		• Use the *slot* argument to specify the slot number in which the network module is installed.
		• Use the *port* argument to specify the port number of the CEM channel to be configured.
		• Use the *channel* argument to specify the CEM channel number to be configured. For a serial channel, enter zero. For a T1 or E1 channel, enter the channel number defined in the **cem-group** command (see the "Creating Circuit Emulation Channels on the T1/E1 Line" section on page 27).

Step 4	**xconnect** *remote-ip-address virtual-connect-ID* **encapsulation** *encapsulation-type*
	
	Example:
	`Router(config-cem)# xconnect 10.2.0.1 0 encapsulation udp`
	`Router(config-cem-xconnect)#`
		Creates one end of a connection between two CEM network modules and enters xconnect configuration mode.
		• Use the *remote-ip-address* argument to specify the IP address of an interface (regular or loopback) on the destination router.
		• Set the *virtual-connect-ID* argument to zero.
		Note Currently the only supported encapsulation type is UDP.

Step 5	**local ip address** *ip-address*
	
	Example:
	`Router(config-cem-xconnect)# local ip address 10.2.0.2`
	`Router(config-cem-xconnect)#`
		Configures the IP address of an interface (regular or loopback) on the source router.
		Note The local IP address must be the same as the remote IP address (at the other end) configured in the **xconnect** command.

Step 6	**local udp port** *port*
	
	Example:
	`Router(config-cem-xconnect)# local udp port 15901`
	`Router(config-cem-xconnect)#`
		Specifies the User Datagram Protocol (UDP) port number of the local CEM channel.
		Note The number of the local UDP port of a CEM channel must be the same as the number of the remote UDP port of the CEM channel at the other end of the connection.

	Command or Action	Purpose

Step 7	**remote udp port** *port*
	
	Example:
	`Router(config-cem-xconnect)# remote udp port 15902`
	`Router(config-cem-xconnect)#`
		Specifies the UDP port number of the remote CEM channel.
		
		Note The number of the remote UDP port of a CEM channel must be the same as the number of the local UDP port of the CEM channel at the other end of the connection.

Step 8	**end**
	
	Example:
	`Router(config-cem-xconnect)# end`
	`Router#`
		Exits xconnect configuration mode and returns to privileged EXEC mode.

Configuring the Circuit Emulation Channel

Perform this task to configure the CEM T1/E1 or serial channel.

This task applies to both the NM-CEM-4TE1 and the NM-CEM-4SER.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cem** *slot*/*port*/*channel*
4. **clock rate** *rate*
5. **clock mode** {**normal** | **split**}
6. **clock source** {**internal** | **loop** | **adaptive**}
7. **payload-size** *size*
8. **dejitter-buffer** *size*
9. **control-lead sampling-rate** *rate*
10. **control-lead state** {**active** | **fail**} *output-lead* {**on** | **off** | **follow**} [{**local** | **remote**} *input-lead*]
11. **data-strobe** *input-lead* {**on** | **off**}
12. **idle-pattern** *pattern length pattern1* [*pattern2*]
13. **failure** {**activation** | **deactivation**} *msec*
14. **signaling** [*on-hook-pattern*] [*off-hook-pattern*] [*msec*]
15. **payload-compression**
16. **data-protection**
17. **ip dscp** [*dscp-value*]
18. **ip tos** *tos*
19. **ip precedence** *precedence*
20. **loopback** {**local** | **network**}
21. **end**
22. **show cem** {*slot*/*port*/*channel* | **summary**}

DETAILED STEPS

	Command or Action	Purpose

Step 1	**enable**
	
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

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

Step 3	**cem** *slot*/*port*/*channel*
	
	Example:
	`Router(config)# cem 3/1/0`
	`Router(config-cem)#`
		Enters CEM configuration mode to configure CEM channels.
		• Use the *slot* argument to specify the slot number in which the network module is installed.
		• Use the *port* argument to specify the port number of the CEM channel to be configured.
		• Use the *channel* argument to specify the CEM channel number to be configured. For a serial channel, enter zero. For a T1 or E1 channel, enter the channel number defined in the **cem-group** command (see the "Creating Circuit Emulation Channels on the T1/E1 Line" section on page 27).

Step 4	**clock rate** *rate*
	
	Example:
	`Router(config-cem)# clock rate 38400`
	`Router(config-cem)#`
		(Optional) For serial channels only. Specifies the nominal bit rate of a serial CEM channel.
		• Use the *rate* argument to specify the data rate of the channel, in bps. Default is 64000.

Step 5	**clock mode** {**normal** | **split**}
	
	Example:
	`Router(config-cem)# clock mode split`
	`Router(config-cem)#`
		(Optional) For serial channels only. Specifies the clock mode of a serial CEM channel.
		• Use the **normal** keyword to specify that the DCE provides both the receive clock and the transmit clock to the attached DTE.
		• Use the **split** keyword to specify that the DCE provides the RxC to the attached DTE and that the DTE provides the external XTC or TT to the DCE.
		Note Depending on the serial cable attached to the port, the port is automatically configured as either a DCE or a DTE.

	Command or Action	Purpose

Step 6	**clock source** {**internal** | **loop** | **adaptive**}
	
	Example:
	`Router(config-cem)# clock source loop`
	`Router(config-cem)#`
		(Optional) Configures the clock source for a serial CEM channel.
		• This step applies only to configuring serial channels. For information about configuring the clock source for T1 or E1 ports, see the "Configuring the T1/E1 Line" section on page 23.
		• Use the **internal** keyword to specify that the clocks provided by the network module to the CPE are derived from the TDM bus backplane clock, if one exists in the router, or from the onboard oscillator on the network module.
		• Use the **loop** keyword to specify that the clock provided by the network module to the attached CPE is derived from the clock received on the same port from the attached CPE.
		• Use the **adaptive** keyword to specify that the clocks provided by the network module to the CPE are locally synthesized based on the average data content of the local dejitter buffer.
		Note The **loop** keyword is valid only when the **clock mode split** command is configured.

Step 7	**payload-size** *size*
	
	Example:
	`Router(config-cem)# payload-size 512`
	`Router(config-cem)#`
		(Optional) Specifies the number of bytes encapsulated into a single IP packet.
		• Use the *size* argument to specify the number of bytes included in the payload of each packet. Default is 32 for a serial CEM channel.
		• For more information about T1 and E1 default values, see the **payload-size** command in the *Cisco IOS Interface and Hardware Component Command Reference*, Release 12.4.

Step 8	**dejitter-buffer** *size*
	
	Example:
	`Router(config-cem)# dejitter-buffer 80`
	`Router(config-cem)#`
		(Optional) Specifies the size of the dejitter buffer used to compensate for the network filter.
		• Use the *size* argument to specify the size of the buffer, in milliseconds. Default is 60.

Step 9	**control-lead sampling-rate** *rate*
	
	Example:
	`Router(config-cem)# control-lead sampling-rate 10`
	`Router(config-cem)#`
		(Optional) For serial channels only. Specifies the sampling rate of input control leads on a serial CEM channel.
		• Use the *rate* argument to specify the frequency with which the control leads are sampled, in samples per second. Default is 0.
		Note Control lead update packets are independent of the data packets from the same channel.

	Command or Action	Purpose

Step 10	**control-lead state** {**active** | **fail**} *output-lead* {**on** | **off** | **follow**} [{**local** | **remote**} *input-lead*]
	
	Example:
	`Router(config-cem)# control-lead state active rts follow remote cts`
	`Router(config-cem)#`
		(Optional) For serial channels only. Specifies the state of each output control lead on a serial CEM channel.
		• Use the **active** keyword to specify the state of the control lead when the connection is active.
		• Use the **fail** keyword to specify the state of the control lead when the connection has failed.
		• Use the *output-lead* argument to specify the name of the control lead.
		• Use the **on** keyword to specify that the control lead is permanently asserted.
		• Use the **off** keyword to specify that the control lead is permanently not asserted.
		• Use the **follow** keyword to specify that the control lead is to follow any changes in the state of an input control lead specified by the **local** or **remote** keywords and the *input-lead* argument.
		• Use the *input-lead* argument to specify the name of the local or remote control lead to follow.
		Note Control lead update packets are independent of the data packets for the same channel.
		Note If the sampling rate is set to 0, sampling is not enabled.

Step 11	**data-strobe** *input-lead* {**on** | **off**}
	
	Example:
	`Router(config-cem)# data-strobe dtr on`
	`Router(config-cem)#`
		(Optional) For serial channels only. Specifies that an input control lead is to be monitored and data is to be packetized and sent only when the specified control lead is in the specified state.
		• Use the *input-lead* argument to specify the input control lead to be monitored to determine whether input data is to be packetized.
		• Use the **on** keyword to specify that data packets are to be sent from this CEM channel only when the specified input lead is asserted.
		• Use the **off** keyword to specify that data packets are to be sent from this CEM channel only when the specified input lead is not asserted.
		• Use this command to save bandwidth when the attached CPE is inactive.
		Note Control lead update packets are still sent even if data packets are withheld.

	Command or Action	Purpose

Step 12	**Cisco NM-CEM-4SER**
	idle-pattern *pattern length pattern1* [*pattern2*]
	
	Cisco NM-CEM-4TE1
	idle-pattern *pattern1*
	
	Example:
	
	Cisco NM-CEM-4SER
	`Router(config-cem)# idle-pattern 53 0x12345678 0x87654321`
	`Router(config-cem)#`
	
	Cisco NM-CEM-4TE1
	`Router(config-cem)# idle-pattern 0x66`
	`Router(config-cem)#`
		(Optional) Defines the idle data pattern to send to the attached CPE when packets are lost or the dejitter buffer experiences an underrun condition.
		
		For serial CEM channels:
		• A bit pattern up to 64 bits long may be specified.
		• Use the *length* argument to specify the total length of the repeating bit pattern. Default is 8 bits.
		• Use the *pattern1* argument to specify up to 32 bits of the least significant bits of the idle data pattern, in hexidecimal notation. Default is 0xFF.
		• Use the *pattern2* argument to specify the most significant bits of the idle data pattern, in hexidecimal notation. If the *length* argument is 32 bits or less, this argument is not permitted.
		
		For T1 or E1 CEM channels:
		• An 8-bit idle data pattern is specified.

Step 13	**failure** {**activation** | **deactivation**} *msec*
	
	Example:
	`Router(config-cem)# failure activation 1000`
	`Router(config-cem)#`
		(Optional) Specifies a time period before a CEM connection enters, or recovers from, a failed state.
		• Use the **activation** keyword to specify how long the software will wait for the detection of a failure of a CEM connection until the CEM channel enters the failed state.
		• Use the **deactivation** keyword to specify how long the software will wait from the detection of a repair to the CEM connection until the CEM channel is returned to an active (up) state.
		• Use the *time* argument to specify the failure activation or deactivation time in milliseconds. The valid range is 50 to 60000. Default is 2000. Any value entered is rounded up to the next multiple of 50 milliseconds.

Step 14	**signaling** [*on-hook-pattern*] [*off-hook-pattern*] [*msec*]
	
	Example:
	`Router(config-cem)# signaling`
	`Router(config-cem)#`
		(Optional) For framed T1 or E1 data channels only. Enables the transport of channel-associated signaling (CAS) bits.

Step 15	**payload-compression**
	
	Example:
	`Router(config-cem)# payload-compression`
	`Router(config-cem)#`
		(Optional) Enables payload compression on a CEM channel.
		
		Note Enabling payload compression adds a delay equal to one packet time.

	Command or Action	Purpose

Step 16	**data-protection**
	
	Example:
	`Router(config-cem)# data-protection`
	`Router(config-cem)#`
		(Optional) Enables data protection by transmitting each data bit twice, once in each of two consecutive data packets.
		• Use the **data-protection** command to protect transmissions from the effects of lost IP packets.
		Caution Use this command carefully because it increases the network bandwidth used by the CEM connection.

Step 17	**ip dscp** [*dscp-value*]
	
	Example:
	`Router(config-cem)# ip dscp 36`
	`Router(config-cem)#`
		(Optional) Configures the IP differentiated services code point (DSCP) for packets originating from this CEM channel.
		• Use the optional *dscp* argument to specify the value placed in the DSCP field of IP packets originating from this channel. Default is 46.
		Note If DSCP is configured, the **ip tos** and **ip precedence** commands are not available because DSCP excludes their use.

Step 18	**ip tos** *tos*
	
	Example:
	`Router(config-cem)# ip tos 11`
	`Router(config-cem)#`
		(Optional) Configures the IP type of service (ToS) bits for the CEM channel.
		• Use the *tos* argument to specify the value placed in the ToS field of IP packets originating from this channel. Default is 5.
		Note If DSCP is configured using the **ip dscp** command, the **ip tos** command is not available because these commands are mutually exclusive.

Step 19	**ip precedence** *precedence*
	
	Example:
	`Router(config-cem)# ip precedence 7`
	`Router(config-cem)#`
		(Optional) Configures the IP precedence bits for the CEM channel.
		• Use the *precedence* argument to specify the value placed in the precedence field of IP packets originating from this channel. Default is 0.
		Note If DSCP is configured using the **ip dscp** command, the **ip precedence** command is not available because these commands are mutually exclusive.

Step 20	**loopback** {**local** | **network**}
	
	Example:
	`Router(config-cem)# loopback network`
	`Router(config-cem)#`
		(Optional) Creates a loopback from a CEM serial channel.
		• Use the **local** keyword to create a loopback for transmitting the information from a locally attached CPE back to the locally attached CPE.
		• Use the **network** keyword to create a loopback for transmitting the data received over the network from a remotely attached CPE back to the remotely attached CPE.
		Note For configuring a loopback on a T1 or E1 port, see the "Configuring the T1/E1 Line" section on page 23.

	Command or Action	Purpose

Step 21	**end**
	
	Example:
	`Router(config-cem)# end`
	`Router#`
		Exits CEM configuration mode and returns to privileged EXEC mode.

Step 22	**show cem** {*slot*/*port*/*channel* | **summary**}
	
	Example:
	`Router# show cem summary`
	`Router#`
		Displays CEM statistics.

Examples

Sample Output for the show cem Command Using the Summary Keyword

The following example shows partial output from the **show cem** command using the **summary** keyword:

```
Router# show cem summary

cem summary

CSTATE: CEM state
LSTATE: line state
OSTATE: operational state
PSIZE: payload-size
PCOMP: payload-compression
DPROT: data-protection

CEM    CSTATE    LSTATE     OSTATE           PSIZE    PCOMP      DPROT
------------------------------------------------------------------------
2/0/0   shutdown   up     config-incomplete   256    disabled   disabled
2/1/0   shutdown   up     config-incomplete   256    disabled   disabled
2/2/0   shutdown   up     config-incomplete   256    disabled   disabled
2/3/0   shutdown   up     config-incomplete   256    disabled   disabled
4/0/1   up         up     active              96     enabled    disabled
4/0/2   up         up     active              96     enabled    disabled
4/0/3   up         up     active              96     enabled    disabled
4/0/4   up         up     active              96     enabled    disabled
4/0/5   up         up     active              96     enabled    disabled
4/0/6   up         up     active              96     enabled    disabled
4/0/7   up         up     active              96     disabled   disabled
4/0/8   up         up     active              96     disabled   disabled
4/0/9   up         up     active              96     disabled   disabled
4/0/10  up         up     active              96     disabled   disabled
```

Sample Output of Basic Configuration of a T1 Network Module to Configure the CEoIP

The following example shows a basic configuration of a T1 network module to configure the CEoIP feature:

```
card type t1 0
controller t1 4/0
 cem-group 6 timeslots 1-4,9,10 speed 64
 framing esf
 linecode b8zs
 clock source adaptive 6
 cablelength long -15db
 crc-threshold 512
```

```
 description T1 line to 3rd floor PBX
 loopback network
 no shutdown
 exit
cem 2/1/6
 xconnect 10.2.0.1 0 encapsulation udp
 local ip address 10.2.0.9
 local udp port 15901
 remote udp port 15902
 payload-size 512
 dejitter-buffer 80
 signaling
exit
```

Sample Output of Serial CEM Network Module

The following example shows a basic configuration of a CEM serial channel to configure the CEoIP feature. Each end of the CEM connection must be configured before the CEM channel is configured.

```
cem 2/0/0
 xconnect 10.3.0.1 0 encapsulation udp
 local ip address 10.3.0.9
 local udp port 15901
 remote udp port 15902
 end
```

Serial CEM Network Module 2

```
cem 2/1/0
 xconnect 10.3.0.9 0 encapsulation udp
 local ip address 10.3.0.1
 local udp port 15902
 remote udp port 15901
 end
```

Serial Channel Configuration

```
cem 2/0/0
 clock rate 38400
 clock mode split
 clock source loop
 payload-size 512
 dejitter-buffer 80
 control-lead sampling-rate 10
 control-lead state active rts follow remote cts
 data-strobe dtr on
 idle-pattern 53 0x12345678 0x87654321
 payload-compression
 data-protection
 ip dscp 36
 loopback network
 end
```

Configuration for the Multiple Service Operator

This section describes how to configure the Cisco cable modem HWIC:

Note This section is for reference only. The cable modem HWIC follows DOCSIS specifications for upgrading the firmware. The cable modem HWIC has its own DOCSIS-compliant software that is upgraded and controlled by the MSO.

How to Download Firmware from the CMTS

This section explains how to download a firmware file from the CMTS to a Cisco cable modem HWIC through the cable modem configuration file for the MSO.

Prerequisites

When using a cable modem configurator tool, you must enable or designate these settings:

- Privacy (optional)
- Upstream and downstream service flows
- Manufacturer code verification certificates (CVC) file

Note The cable modem verifies the manufacturer's digital signature and, if present, the cable operator's digital signature, using the CVCs provided in the DOCSIS configuration file. If the signatures are valid, the cable modem loads and runs the software.

- Network access enabled
- Firmware filename
- Provisioning server IP address
- TLV 11 ODI (docDevSwAdminStatus.0)

Note TLV11 specifies the value of the docsDevSwAdminStatus.0 MIB object. This MIB object has an equivalent integral value of 1.3.6.1.2.1.69.1.3.3.0. You can specify either the MIB object or its equivalent value. When set to upgradeFromMgt(1) or equivalent integral value of 2, the cable modem initiates a TFTP firmware download using docsDevSwFilename MIB object. When this object is used, the router automatically resets itself after the firmware download is complete. When set to allowProvisioningUpgrade(2), or equivalent integral value of 2, the cable modem uses the software version information supplied by the provisioning server after you reboot the system. The provisioning server is provided by your MSO.

Preparing the Cable Modem Configuration File

To prepare the cable modem configuration file, follow these steps:

Step 1 Prepare the cable modem configuration file using a cable modem configuration editor (such as Cisco Configuration Editor).

Caution The following example is for reference only. This example provides only the minimum number of fields required to initiate a firmware download. Specific values based on your configuration needs should replace the values below.

```
FileVersion = Version 5.0

03 (Net Access Control)       = 1
09 (Software Upgrade File)    = C21031012eFU02172006.CDF
11 (MIB Object)                 = 30 12 06 0a 2b 06 01 02 01 45 01 03 03 00 02 04 00 00 00
02

17 (BLP Config Settings)
   S01 (Author Timeout)       = 5
   S02 (Re-auth Wait Timeout) = 5
   S03 (Author Wait Timeout)  = 60
   S04 (Oper Wait Timeout)    = 2
   S05 (Re-Key Wait Timeout)  = 2
   S06 (TEK Grace Time)       = 60
   S07 (Auth Rej Wait Timeout)= 5
   S08 (SA Map Wait Timeout)  = 5

21 (Software Upgrade Addr)    = 007.000.000.001

32 (Manufacturer CVC)           = us_cvc_cert.crt

18 (Maximum Number of CPE)    = 10
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference)     = 1
   S06 (QoS Parameter Set Type)     = 7
   S08 (Max Sustained Traffic Rate) = 20000000
   S15 (Service Flow Sched Type)    = 2

25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)     = 5
   S06 (QoS Parameter Set Type)     = 7
   S08 (Max Sustained Traffic Rate) = 20000000

29 (Privacy Enable)           = 0
```

Step 2 Download the prepared cable modem configuration file to the CMTS bootflash using the **copy tftp: bootflash** command.

Step 3 Download the desired firmware image to the CMTS bootflash using the **copy tftp: bootflash** command.

Step 4 Use the **tftp-server** command to prepare the cable modem configuration file prepared in Step 1 and to configure the lines in the CMTS configuration mode to prepare for downloading the firmware:

```
Router(config)# tftp-server firmware-name
Router(config)# tftp-server bootflash:C21031013cFU04072006.CDF

Router(config)# tftp-server Cable-Modem config-file
Router(config)# tftp-server bootflash:00dd_2bbo_695a.bin
```

Step 5 Reset the cable router with CMTS commands. This downloads a new cable modem configuration file to the cable router. This step also downloads a new firmware version onto the cable router since the new firmware filename is specified in the cable modem configuration file. **00d0.2bfe.66ce** represents the MAC address of the cable modem installed in the router. For example:

```
ats1-cmts-1# clear cable modem 00d0.2bfe.66ce reset
```

Step 6 Verify the firmware download procedure was successful by using the **show controller cable status** command:

```
Router# show controller port-number status
Router# show controller cable 0/0/1 status
```

Note This procedure takes approximately 2 to 3 minutes to complete.

The following example shows the output from this command. The new firmware version is displayed in the *Software Hidden version* line of text.

```
Router# show controller cable 0 status
 Cable Modem Information:
   Software version           2.103.1012
   Software Hidden version    2.103.1012e
   Hardware version
   Cable IP address           7.0.0.23/24
   DOCSIS mode                3 (2_0)
   BPI status                 1 (DISABLED)
   Uptime (seconds)           170871
   Current state              16 (OPERATIONAL)
   Cable MAC address          00d0.2bfe.66ce
   Internal MAC address       00d0.2bfe.66cf
   Internal IP address        192.168.100.1/24
   Downstream buffers free    128
   Downstream buffers used    0
   Upstream buffers free      254
   Upstream buffers used      0
   MAC SDRAM free (Kbytes)    20361008
   MAC SDRAM used (Kbytes)    7563552
   MAC Flash free (Kbytes)    1823657
   MAC Flash used (Kbytes)    2337879
```

Note If your cable modem configuration file has Baseline Privacy Interface (BPI or BPI+) enabled, perform the following steps on the CMTS:

1. Configure the date and time on the router in privileged EXEC mode:

```
CMTS# clock set hh:mm:ss day month year
CMTS# clock set 12:22:36 23 July 2006

CMTS# clock update-calendar
CMTS# clock update-calendar
```

2. Download the DOCSIS root certificate file onto the CMTS bootflash.

To download the DOCSIS root certificate to the CMTS, see *Downloading the DOCSIS Root Certificate to the CMTS* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/cable/cab_rout/cmtsfg/ufg_docs.htm#wp1217174

Vendor Specific Type-Length-Values 42

Use the Cisco vendor-specific Type-Length-Value (TLVs) 42 in the text file to convert to binary in the configuration file.

TLV 42

Use TLV 42 to enable downstream routing protocols using multicast address such as RIPv2.

To support DOCSIS configuration file-based enabling downstream routing protocols using multicast address such as RIPv2, your DOCSIS configuration file editor must support the inclusion of the Cisco vendor-specific TLV 42.

The following example shows how to use TLV 42 in the configuration file. This information can be found in the vendor information specific field (VISF):

```
00 (Multicast Mac Address)              = 42 %hex 01 00 5e 00 00 09
```

Note The value of TLV 42 is fixed for all routers.

Additional References

Related Documents

Related Topic	Document Title
Hardware installation instructions for interface cards	*Cisco Interface Cards Installation Guide*
Configuration fundamentals for Cisco IOS software release 12.4	*Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4*
DOCSIS 2.0 specifications	*Data-Over-Cable Service Interface Specifications, DOCSIS 2.0: Radio Frequency Interface Specification*
Configuration information for configuring NAT	*Configuring NAT*
Configuration information for configuring DHCP	*Configuring DHCP*
Configuration information for configuring Easy VPN	*Configuration Example: Easy VPN*
Configuration information for configuring IGMP	*UDLR Tunnel ARP and IGMP Proxy*
Configuration information for configuring CoIP	*Circuit Emulation over IP*

MIBs

MIBs	MIBs Link
CISCO-SMI-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
CISCO-STACK-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
CISCO-VTP-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
CPU-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
DOCSIS-BPI2-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
DOCSIS-IF-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
DOCSIS-IFEXT2-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
DOCSIS-CABLE-DEVICE-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
DOCSIS-CABLE-DEVICE-TRAP-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
DOCSIS-QOS-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
ENTITY-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
FDDI-SMT73-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

MIBs	MIBs Link
IANiftype-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
OLD-CISCO-CPU-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
IF-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
RFI-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
RMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
SNMP-FRAMEWORK-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
SNMPv2-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
SNMPv2-CONF-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
SNMPv2-TC-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
SNMPv2-TC-v1-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
SNMPv2-SMI-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs
TOKEN-RING-RMON-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1155	*Structure and identification of management information for TCP/IP-based internets*
RFC 1212	*Concise MIB definitions*
RFC 1213	*Management Information Base for Network Management of TCP/IP-based internets:MIB-II*
RFC 1215	*Convention for defining traps for use with the SNMP*
RFC 1271	*Remote Network Monitoring Management Information Base*
RFC 1493	*Definitions of Managed Objects for Bridges*
RFC 2011	*SNMPv2 Management Information Base for the Internet Protocol using SMIv2*
RFC 2013	*SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2*
RFC 2576	*Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*
RFC 2665	*Definitions of Managed Objects for the Ethernet-like Interface Types Base*
RFC 2669	*DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems*
RFC 2670	*Radio Frequency (RF) Interface Management Information Base for DOCSIS compliant RF interfaces*
RFC 2786	*Diffie-Helman USM Key Management Information Base and Textual Convention*
RFC 2863	*The Interfaces Group MIB*
RFC 2933	*Internet Group Management Protocol MIB*
RFC 3083	*Baseline Privacy Interface Management Information Base for DOCSIS Compliant Cable Modems and Cable Modem Termination Systems*
RFC 3410	*Introduction and Applicability Statements for Internet-Standard Management Framework*
RFC 3411	*An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*
RFC 3412	*Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*
RFC 3413	*Simple Network Management Protocol (SNMP) Applications*
RFC 3414	*User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*
RFC 3415	*View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*
RFC 3418	*Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*
RFC 4131	*Management Information Base for Data Over Cable Service Interface Specification (DOCSIS) Cable Modems and Cable Modem Termination Systems for Baseline Privacy Plus*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Commands at a Glance

This section documents new and modified commands only.

New Commands

Modified Commands

clear interface cable-modem

To reset the controller for a specified cable modem daughter card, use the **clear interface cable-modem** command in privileged EXEC mode.

clear interface cable-modem

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(6)XC	This command was introduced

Usage Guidelines

Use this command as an alternative to the cable-modem power cycle command.

Examples

The following example clears the interface on the selected slot and port:

```
*May 17 16:36:57.344: %CABLE_MODEM_HWIC-6-RESET: Interface Cable-Modem0/2/0 has been
reset: clear command
*May 17 16:37:05.348: %LINK-3-UPDOWN: Interface Cable-Modem0/2/0, changed state to down
*May 17 16:37:06.348: %LINEPROTO-5-UPDOWN: Line protocol on Interface Cable-Modem0/2/0,
changed state to down
*May 17 16:37:19.740: %LINK-3-UPDOWN: Interface Cable-Modem0/2/0, changed state to up
*May 17 16:37:27.996: %LINEPROTO-5-UPDOWN: Line protocol on Interface Cable-Modem0/2/0,
changed state to up
```

Related Commands

Command	Description
show interfaces	Displays statistics for all interfaces configured.
show interfaces cable-modem	Displays statistics for all interfaces configured on the port.

debug cable-modem driver

To enable debugging on the WIC and HWIC driver, use the **debug cable-modem driver** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cable-modem driver [**detail | error**]

no debug cable-modem driver [**detail | error**]

Syntax Description

detail	(Optional) Provides additional detailed debugging information.
error	(Optional) Enables driver debugging of the driver error paths.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(6)XC	This command was introduced.

Usage Guidelines

Significant errors are acknowledged by counters or error messages. Error debugging provides more detailed information.

Debugging can also be enabled or disabled by using the **debug condition interface cable-modem** *port* command. If a condition interface is enabled for one port, the debugging capability is disabled for the remaining ports.

Examples

The following example turns CM driver debugging on:

```
Router# debug cable-modem driver

CM driver debugging is on
```

Related Commands

Command	Description
debug condition interface cable-modem *port*	Enables debugging messages for additional interfaces.

debug cable-modem rbcp

To activate debugging on the modem router blade control port (RBCP) code, use the **debug cable-modem rbcp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cable-modem rbcp {**events** | **messages** | **states**}

Syntax Description

events	Enables the RBCP finite state machine internal event debugging.
messages	Enables the RBCP message debugging. When enabled, a debug message is generated for every received and sent RBCP request and response.
states	Enables the RBCP finite state machine state transition debugging.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(6)XC	This command was introduced.

Usage Guidelines

When no keyword is selected, this command enables miscellaneous RBCP debugging.

When the **messages** keyword is enabled, a debug message is generated for every RBCP request and response.

Cisco IOS software RBCP support also contains its own debug facility with the **debug scp data** and **debug scp packets** commands.

Debugging can be also be enabled or disabled by using the **debug condition interface cable-modem** *port* command. If a condition is enabled for one port, the debugging capability is disabled for the remaining ports.

Examples

```
Router# debug cable-modem rbcp messages

CM rbcp messages debugging is on
```

Related Commands

Command	Description
debug condition interface cable-modem *port*	Enables debugging messages for additional interfaces.
debug scp data	Displays SCP data information.
debug scp packets	Displays SCP header information.

debug cable-modem startup

To enable modem initialization code debugging, use the **debug cable-modem startup** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cable-modem startup

no debug cable-modem startup

Syntax Description

This command has no arguments or keywords.

Defaults

This command is disabled by default.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(6)XC	This command was introduced.

Usage Guidelines

Debugging can also be enabled or disabled by using the **debug condition interface cable-modem** *port* command. If a condition is enabled for one port, the debugging capability is disabled for the remaining ports.

Examples

The following shows CM startup debugging turned on:

```
Router# debug cable-modem startup

CM startup debugging is on
```

Related Commands

Command	Description
debug condition interface cable-modem *port*	Enables debugging messages for additional interfaces.

service-flow primary upstream

To assign a QoS policy to the data traveling between the cable modem to the multiple service operator (MSO) cable modem termination system (CMTS), use the **service-flow primary upstream** command in interface configuration mode. To disable, use the **no** form of this command.

service-flow primary upstream

no service-flow primary upstream

Syntax Description

This command has no arguments or keywords.

Defaults

This command is disabled by default.

Command Modes

Interface configuration

Command History

Release	Modification
12.4(6)XC	This command was introduced.

Usage Guidelines

This command is supported in the upstream direction only so only the output form of the command is available. Service flows are unidirectional.

Examples

The following example assigns a QoS policy to the data traveling between the cable modem to the MSO CMTS:

```
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)# interface Cable-Modem 0/2/0
Router(config-if)# service-flow primary upstream

Router(config-serviceflow)#
```

service-module ip address

To define the IP address for the internal network module-side interface on a content engine network module (NM-CE-BP) or Cisco IP cable modem interface satellite WAN network module, use the **service-module ip address** command in content-engine interface configuration mode or satellite interface configuration mode. To delete the IP address associated with this interface, use the **no** form of this command.

service-module ip address {*nm-side-ip-addr subnet-mask*}

no service-module ip address

Syntax Description

nm-side-ip-addr	IP address of the internal network-module-side interface on a content engine (CE) network module (NM-CE-BP) or Cisco IP cable modem interface satellite WAN network module.
subnet-mask	Subnet mask to append to the IP address.

Defaults

No default behavior or values

Command Modes

Content-engine interface configuration
Satellite interface configuration

Command History

Release	Modification
12.2(11)YT	This command was introduced for the CE network module.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.3(14)T	This command was implemented on the Cisco IP VSAT satellite WAN network module (NM-1VSAT-GILAT).
12.4(6)XC	This command was modified with the *subnet-mask* argument.

Usage Guidelines

For the NM-1VSAT-GILAT network module, the **service-module ip address** command is typically not used. The NM-1VSAT-GILAT network module IP address is automatically configured when you enter the **ip address** command in satellite interface configuration mode to configure the IP address and subnet mask of the router satellite interface with the following conditions:

- The IP address leaves a remainder of 2 when the last octet is divided by 4.
- The subnet mask has /30 or fewer masking bits.

If you use this method to configure the IP address for the router satellite interface, the system automatically configures the IP address and subnet mask on the NM-1VSAT-GILAT network module with these results:

- The IP address is 1 less than the IP address you configured for the router satellite interface.
- The subnet mask is /30.

You can override the automatically configured IP address and mask by manually entering the **service-module ip address** command.

Note The automatically configured IP address does not appear in the router configuration, because the **service-module ip address** command is considered to be set to its default value. Similarly, if you manually configure an IP address and subnet mask that are identical to the automatically configured IP address and subnet mask, the **service-module ip address** command does *not* appear in the router configuration.

Examples

The following example shows how to define an IP address for the internal network-module-side interface on the content engine network module in slot 1:

```
Router(config)# interface content-engine 1/0
Router(config-if)# service-module ip address 172.18.12.26 255.255.255.0
Router(config-if)# exit
```

In the following example, the router satellite interface is assigned an IP address (10.0.0.7), the last octet of which does *not* leave a remainder of 2 when divided by 4. The system displays a message to manually configure the IP address for the NM-1VSAT-GILAT network module. Notice that the IP addresses for both the router satellite interface and the NM-1VSAT-GILAT network module appear in the running configuration.

```
Router(config)# interface satellite 1/0
Router(config-if)# ip address 10.0.0.7 255.255.255.0

%VSAT-6-PIMINCOMPADDR: The IP address configured on Satellite1/0
      requires a manually configured IP address for the satellite module

Router(config-if)# service-module ip address 10.0.0.6 255.255.255.0
Router(config-if)# end
Router# show running-config | begin Satellite

interface Satellite 1/0
   ip address 10.0.0.7 255.255.255.0
   service-module ip address 10.0.0.6 255.255.255.0
.
.
.
```

In the following example, the router satellite interface IP address is configured as 10.0.0.6. Because the last octet of the IP address leaves a remainder of 2 when divided by 4, the system automatically configures the IP address for the NM-1VSAT-GILAT network module.

Although the NM-1VSAT-GILAT network module IP address and mask do not appear in the router configuration, you know that the IP address is 1 less than the IP address of the router satellite interface and has a subnet mask of /30. In this case, the NM-1VSAT-GILAT network module is automatically configured with the following IP address and mask: 10.0.0.5 255.255.255.252.

```
!
interface Satellite 1/0
 ip address 10.0.0.6 255.255.255.0
!
```

In the following example, the router satellite interface IP address is configured as 10.0.0.6. Because the last octet of the IP address leaves a remainder of 2 when divided by 4, the system automatically configures the IP address and mask for the NM-1VSAT-GILAT network module as 10.0.0.5 255.255.255.252.

Nevertheless, the NM-1VSAT-GILAT network module IP address and mask are manually configured as 10.0.0.1 255.255.255.0 to override the automatically derived IP address and mask. Notice that the IP addresses for both the router satellite interface and the NM-1VSAT-GILAT network module appear in the running configuration.

```
!
interface Satellite 1/0
 ip address 10.0.0.6 255.255.255.0
 service-module ip address 10.0.0.1 255.255.255.0
!
```

Related Commands

Command	Description
show controllers content-engine	Displays controller information for CE network modules.
show controllers satellite	Displays controller information about the internal router interface that connects to an installed Cisco IP VSAT satellite WAN network module (NM-1VSAT-GILAT).
show interfaces satellite	Displays general interface settings and traffic rates for the internal router interface that connects to an installed Cisco IP VSAT satellite WAN network module (NM-1VSAT-GILAT).
show interfaces content-engine	Displays basic interface configuration information for a CE network module.

show controllers cable-modem

To display status information for the router, use the **show controllers cable-modem** *port* command in privileged EXEC mode.

show controllers cable-modem *port* [**all | classifiers | cm-cert | crypto des | filters | internal-mac | lookup-table | mac** {**counts crashdump | hardware | log | state**} **| manuf-cert | phy | service-flows | status | tuner**]

Syntax Description

port	Selects the port.
all	(Optional) Displays all of the controller information for the given port.
classifiers	(Optional) Displays the DOCSIS 1.1/2.0 packet classifiers currently in use on the router.
cm-cert	(Optional) Displays the cable modem public key X.509 certificate.
crypto-des	(Optional) Displays the DOCSIS Data Encryption Standard settings for the port.
filters	(Optional) Displays the DOCSIS filters that are enabled on the port for filtering received frames.
internal mac	(Optional) Displays the settings for the internal WIC or WHIC interface information. These settings include information for the MII interface between the interface card and the daughter card.
counts	Displays the Hybrid Fiber Coax (HFC) statistics.
crashdump	Displays the most recent daughter card crashdump information. The daughter card must be running to have the crashdump information read.
hardware	Displays the Broadcom registers and hardware queues.
log	Displays the MAC log messages (up to 1023 entries).
state	Displays the MAC state information such as downstream and upstream frequencies, symbol rates, mini-slot size, and burst descriptor.
manuf-cert	(Optional) Displays the manufacturer's X.509 certificate.
phy	(Optional) Displays information about the cable modem's physical interface.
service-flows	(Optional) Displays detailed information about the service flows that are configured on this port. This command does not support the "summary" sub-command or *sfid* argument.
status	(Optional) Displays status information about the firmware.
tuner	(Optional) Displays the settings for the upstream and downstream tuners used by the cable interface.

Command Default

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(6)XC	This command was introduced.

Examples

The following example displays information about the cable modem physical interface:

```
Router# show controllers cable-modem 1 phy

Phy Minislots to MAC Bytes table for kLongDataGrantIUC
MAC Bytes for (Mslot 10's + offset)
Mslot Mslot offset
10's 0 1 2 3 4 5 6 7 8 9
===== ===== ===== ===== ===== ===== ===== ===== ===== ===== =====
0 0 0 0 0 0 0 0 0 213 231
1 261 293 325 357 389 421 453 462 501 533
2 565 597 629 661 693 709 741 773 805 837
3 869 901 924 949 981 1013 1045 1077 1109 1141
4 1155 1189 1221 1253 1285 1317 1349 1381 1386 1429
5 1461 1493 1525 1557 1589 1617 1637 1669 1701 1733
6 1765 1797 1829 1848 1877 1909 1941 1973 2005 <-- max burst
Request Opportunity Burst Size (Mslots) = 2
Initial Ranging Opportunity Burst Size (Mslots) =
Phy Burst Size (Mslots) to send (1) MAC byte for
Std Short grant = 2
Std Long grant =
```

The following example displays firmware status information:

```
Router# show controllers cable-modem 1 status

Cable Modem Information:
Software version 2.103.1003
Software Hidden version 2.01
Hardware version 2.103.1003a
Cable IP address 0.0.0.0/0
DOCSIS mode 0 (UNKNOWN)
BPI status 1 (DISABLED)
Uptime (seconds) 0
Current state 2 (NOT_SYNCHRONIZED)
Cable MAC address 00d0.59e1.03fe
Internal MAC address 00d0.59e1.03ff
Internal IP address 0.0.0.0/0
Downstream buffers free 128
Downstream buffers used 0
Upstream buffers free 255
Upstream buffers used 255
MAC SDRAM free (Kbytes) 255
MAC SDRAM used (Kbytes) 255
MAC Flash free (Kbytes) 255
MAC Flash used (Kbytes) 255
```

show interfaces cable-modem

To display statistics for all interfaces configured on the port, use the **show interfaces cable-modem** in privileged EXEC mode.

show interfaces cable-modem *port*

Syntax Description

port	The port number.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(6)XC	This command was introduced.

Usage Guidelines

The resulting output varies, depending on the network for which an interface has been configured.

Examples

The following example shows the HFC state on the modem:

```
c2801-61# show interfaces Cable-Modem 0/1/0
cable-modem0/1/0 is up, line protocol is up
  HFC state is OPERATIONAL, HFC MAC address is 00d0.59e1.2073
  Hardware is Cable modem, address is 0014.f26d.10b2 (bia 0014.f26d.10b2)
  Internet address is 12.0.0.61/8
  MTU 1500 bytes, BW 1544 Kbit, DLY 6470 usec,
     reliability 255/255, txload 247/255, rxload 246/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:00, output hang never
  Last clearing of "show interface" counters 00:07:03
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 83594
  Queueing strategy: Class-based queueing
  Output queue: 61/1000/64/83594 (size/max total/threshold/drops)
     Conversations  2/5/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 232 kilobits/sec
  30 second input rate 2581000 bits/sec, 987 packets/sec
  30 second output rate 1585000 bits/sec, 639 packets/sec
  HFC input: 0 errors, 0 discards, 0 unknown protocols 0 flow control discards
  HFC output: 0 errors, 0 discards
     304582 packets input, 105339474 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 1 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     228195 packets output, 78392605 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

This following table describes the fields within the HFC state (the DOCSIS state for the cable modem connection to the CMTS.

Table 3 show insterfaces cable-modem Field Description

HFC State Values	Description
NOT_READY	Cable modem controller is resetting.
NOT_SYNCHRONIZED	Cable modem controller is starting the downstream frequency scan.
PHY_SYNCHRONIZED	Cable modem controller locked the downstream signal and is collecting the upstream channel parameter information.
US_PARAMETERS_ACQUIRED	Cable modem controller collected upstream channel parameter information and is trying to lock upstream frequency.
RANGING_COMPLETE	Cable modem controller received the CMTS range response, has finished downstream/upstream lock process, and is initializing IP.
IP_COMPLETE	Cable modem controller has the IP information.
WAITING_FOR_DHCP_OFFER	Cable modem controller is sending DHCP request to the CMTS.
WAITING_FOR_DHCP_RESPONSE	Cable modem controller is waiting for DHCP response from the CMTS.
WAITING_FOR_TIME_SERVER	Cable modem controller is starting the ToD service.
TOD_ESTABLISHED	Cable modem controller has received the ToD packet and has synchronized its local time.
WAITING_FOR_TFTP	Cable modem controller is downloading its running configuration from the CMTS-defined TFTP server.
PARAM_TRANSFER_COMPLETE	Cable modem controller has completed transferring its running configuration.
REGISTRATION_COMPLETE	Cable modem controller sends out its registration request, and CMTS has accepted it.
REFUSED_BY_CMTS	Cable modem controller's registration request has been rejected by CMTS.
FORWARDING_DENIED	The cable modem's controller's registration to CMTS was successful, but network access is disabled in running configuration.
OPERATIONAL	The cable modem controller is ready for service.
UNKNOWN	The cable modem controller is an undefined state

The following table describes input error descriptions.

Table 4 Input Error Description

Input Error	Description
errors	The total number of input packets discarded on the cable modem controller.
discards	The number of input packets discarded due to a momentary lack of resources.
unknown protocols	The number of input packets discarded because they have unsupported or unknown protocol values.
flow control discards	The number of input packets discarded because the cable modem controller overflowed transferring packets to the router.

The following table describes output error descriptions.

Table 5 Output Error Description

Output Error	Description
errors	Total number of output packets discarded on the cable modem controller.
discards	Total number of output packets discarded due to a momentary lack of resources.

Related Commands

Command	Description
show interfaces	Displays statistics for all interfaces.
show interfaces cable-modem	Displays statistics for all interfaces configured on the port.

show ip access-list

To display the contents of all current IP access lists, use the **show ip access-list** command in user EXEC or privileged EXEC mode.

show ip access-list [*access-list-number* | *access-list-name* | **dynamic** *access-list-name* | **interface** *interface-name* [**in** | **out**]]

Syntax Description

access-list-number	(Optional) Number of the IP access list to display.
access-list-name	(Optional) Name of the IP access list to display.
dynamic *access-list-name*	(Optional) Displays the specified dynamic IP access lists.
interface *interface-name*	(Optional) Displays the access list for the specified interface.
in	(Optional) Displays input interface statistics.
out	(Optional) Displays output interface statistics.

Defaults

All standard and extended IP access lists are displayed.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
10.3	This command was introduced.
12.3(7)T	The **dynamic** keyword was added.
12.4(6)T	The **interface** *interface-name* keyword/attribute pair was added. The **in** and **out** keywords were added.
12.4(6)XC	Additional example output using the **dynamic** keyword was added.

Usage Guidelines

The **show ip access-list** command provides output identical to the **show access-lists** command, except that the first command is IP-specific and allows you to specify a particular access list.

Examples

The following is sample output from the **show ip access-list** command when all access lists are requested:

```
Router# show ip access-list

Extended IP access list 101
   deny udp any any eq ntp
   permit tcp any any
   permit udp any any eq tftp
   permit icmp any any
   permit udp any any eq domain
```

The following is sample output from the **show ip access-list** command when the name of a specific access list is requested:

```
Router# show ip access-list Internetfilter

Extended IP access list Internetfilter
   permit tcp any 10.31.0.0 0.0.255.255 eq telnet
   deny tcp any any
   deny udp any 10.31.0.0 0.0.255.255 lt 1024
   deny ip any any log
```

The following is sample output from the **show ip access-list** command, which shows input statistics for FastEthernet interface 0/0:

```
Router# show ip access-list interface FastEthernet 0/0 in

Extended IP access list 150 in
   10 permit ip host 10.1.1.1 any
   30 permit ip host 20.2.2.2 any (15 matches)
```

The following is sample output from the **show ip access-list** command using the **dynamic** keyword:

```
show ip access-lists dynamic
Extended IP access list CM_SF#1
    10 permit udp any any eq 5060 (650 matches)
    20 permit tcp any any eq 5060
    30 permit udp any any dscp ef (806184 matches) c2801-61#
```

To check your configuration when the **dynamic** keyword is used, use the **show run interfaces cable** command:

```
show run interfaces cable 0/1/0
Building configuration...

Current configuration : 144 bytes
!
interface cable-modem0/1/0
 ip address dhcp
 load-interval 30
 no keepalive
  service-flow primary upstream
   service-policy output llq
end

c2801-61#
```

Related Commands

Command	Description
show interfaces	Displays statistics for all interfaces.
show interfaces cable-modem	Displays statistics for all interfaces configured on the port.
show run interfaces cable	Displays statistics on the cable modem.