Cisco 800M Series Integrated Services Routers Software Configuration Guide

June 8, 2015

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

Cisco 800M Series, Integrated Services Routers Software Configuration Guide

Preface

This preface describes the objectives, audience, organization, conventions of this guide, and the references that accompany this document set. The following sections are provided:

Objectives

This guide provides information about how to configure the various features of Cisco 800M Series integrated services routers (ISRs).

Audience

This document is written for experienced technical workers who install, monitor, and troubleshoot routers under a service contract, or who work for an information technology (IT) department.

Organization

This document is organized into the following chapters:

Chapter	Description
Overview	Provides an overivew of the hardware and software features of Cisco 800M Series ISRs.
Basic Router Configuration	Describes how to perform the basic router configuration, interface configuration, and routing configuration.
Configuring 3G Wireless WAN	Describes the configuration procedures for 3G Wireless WAN module on the Cisco 800M Series ISR.

Chapter	Description
Configuring the Serial Interface	Describes how to configure the serial module on the Cisco 800M Series ISR.
Configuring Ethernet Switch Ports	Provides an overview of the configuration tasks for the Gigabit Ethernet switch on the Cisco 800M Series ISR.
Configuring Security Features	Describes how to configure security features for the Cisco 800M Series ISR.
Configuring QoS	Describes configuring the Quality of Service(QoS) features supported on the Cisco 800M Series ISR.
Configuring Network Management Features	Describes configuring the network management features for the Cisco 800M Series ISR.
Configuring IP Addressing and IP Services Features	Describes configuring IP addressing and IP services features for the Cisco 800M Series ISR.

Conventions

This document uses the following conventions:

Convention	Indication
bold font	Commands and keywords and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`courier` font	Terminal sessions and information the system displays appear in `courier` font.
< >	Non-printing characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means *reader take note*.

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

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

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

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

Related Documentation

In addition to the Cisco 800M Series ISR Software Configuration Guide (this document), the following reference guides are included:

Type of Document	Links
Hardware	*Cisco 800M Series Routers Hardware Installation Guide*
Regulatory Compliance	*Regulatory Compliance and Safety Information for Cisco 800 Series Routers*

Obtaining Documentation and Submitting a Service Request

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

http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html

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

Cisco 800M Series Integrated Services Routers Overview

This chapter provides an overview of Cisco 800M Series integrated services routers (ISRs), and information about how to configure the features and contains the following sections:

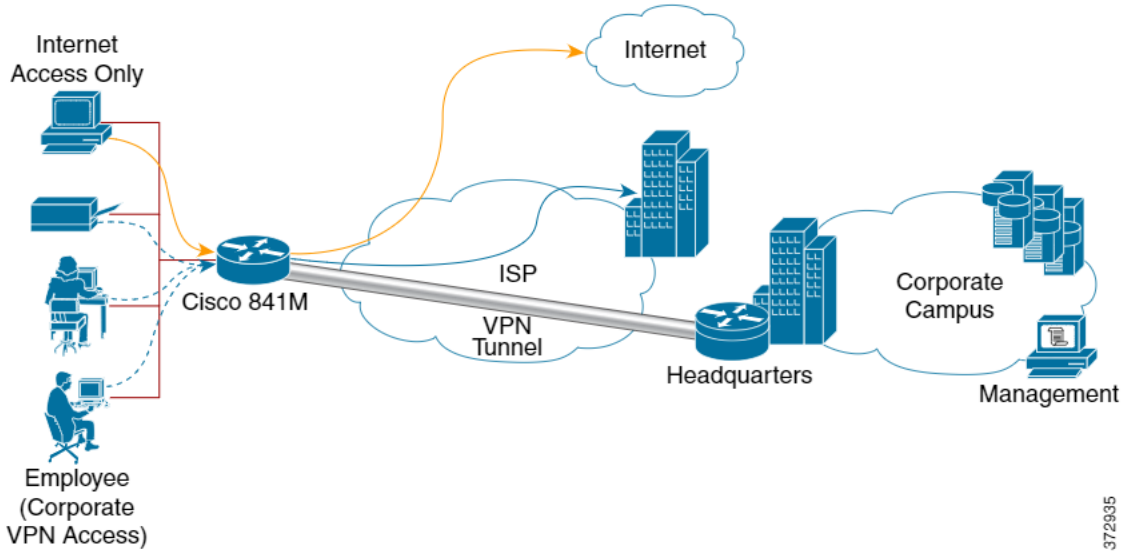
Overview of the Cisco 800M Series ISR

Cisco 800M Series ISRs are entry level branch routers that provide secure network connectivity for small offices to a central location. Cisco 800M Series ISRs are modular routers and provides flexibleWAN connectivity options including Gigabit Ethernet (GE), Serial, and 3G to connect the branch office to central office over a secure tunnel. The Cisco 800M Series ISR can be used for deployment in remote small offices, ATMs and retail stores.

The Cisco 800M Series ISR runs Cisco IOS Software and delivers built-in security in a single software image without any additional software license.The Cisco 800M Series ISR provides an open, extensible environment for developing and hosting applications at the network edge.

Figure 1-1 explains a scenario where the Cisco 800M Series ISR is deployed to provide remote connectivity from a small office to central office over secure VPN tunnels. In this scenario corporate users use a separate VLAN than the Internet users.

Figure 1-1 Cisco 800M Series Deployment Example



Cisco 800M Series ISR Models

Cisco 800M Series ISRs are available in two models:

- Cisco C841M-4X (Integrated 4 port GE LAN and 2 port GE WAN)
- Cisco C841M-8X (Integrated 8 port GE LAN and 2 port GE WAN)

The Cisco 800M Series ISR supports highly available and redundant WAN connection options and allows you to easily migrate to different WAN connections. The Cisco 800M Series ISR has 2 WAN slots that can host single port serial module or multi-mode 3G Wireless WAN module. The 3G Wireless WAN module supports multiple 3G technologies on the same pluggable WAN interface and provides service provider independence. These modules are field replaceable which provides flexibility and ease of procurement.

Table 1-1 describes the pluggable WAN configuration modules available for Cisco 800M Series ISR models.

Table 1-1 Pluggable WAN Modules

Module	Description
WIM-3G	Multimode 3G Wireless WAN module
WIM-1T	Single port serial module

The combination of WAN modules supported by the Cisco 800M Series ISR is given as follows:

- Multimode 3G Wireless WAN module in slot 0 and single port serial module in slot 1
- Single port serial module in slot 0 and multimode 3G Wireless WAN module in slot 1
- Single port serial module in slot 0 and single port serial module in slot 1

Note Only one 3G Wireless WAN module is supported at a time on the Cisco 800M Series ISR. If two 3G Wireless WAN modules are present in the 800M Series ISR, the 3G Wireless WAN module in the second slot will be powered down.

Table 1-2 summarizes the LAN and WAN interface options available for the Cisco 800M Series ISR models.

Table 1-2 LAN and WAN Interfaces of the Cisco 800M Series ISRs

800M Series Models	LAN Interfaces	GE WAN Interfaces	Serial WAN	3G WAN
Cisco C841M-4X	4 Gigabit Ethernet LAN ports	2 Gigabit Ethernet ports	Single port serial	Multimode 3G (GSM/CDMA)
Cisco C841M-8X	8 Gigabit Ethernet LAN ports	2 Gigabit Ethernet ports	Single port serial	Multimode 3G (GSM/CDMA)

Cisco 800M Series ISR Features

The Cisco 800M series ISR comes with advanced IP Services license and support the features available for advanced IP services license.

Some of the key features supported by Cisco 800M Series ISRs are listed as follows:

- GSM/CDMA 3G Wirless WAN modes
- Serial WAN link with HDLC, PPP and Frame Relay encapsulations
- Advanced security features including IP Security (IPsec) VPNs, Dynamic Multipoint VPN (DMVPN) andTunnel-less Group Encrypted Transport (GETVPN)
- Layer 2 features including VLAN/802.1q Trunking
- Integrated device management using Cisco Configuration Professional Express.
- Remote management and network monitoring using SNMP, Telnet, and HTTP, and locally through a console port

LEDs on the Cisco 800M Series ISR

Table 1-3 describes the LEDs on the Cisco 800M Series ISR.

Table 1-3 LEDs on the Cisco 800M Series ISR

LED	Color	Description
SYS	Green (blinking)	System is booting.
	OFF	System is off.
	Solid Green	System is in active status.
Slot 0	Amber	Module is inserted in slot 0 but the module is not in service.
	Green	Module is inserted in slot 0 and the module is in service.
	OFF	No module is present in slot 0.
Slot 1	Amber	Module is inserted in slot 1 but the module is not in service.
	Green	Module is inserted in slot 1 and the module is in service.
	OFF	No module is present in slot 1.
VPN OK	Green	At least one VPN session is active.
	OFF	VPN not connected.
PPP OK	Green	At least one PPP session is active.
	OFF	PPP session is not connected.
LAN	Solid Green	LAN connection is established
	Green (Blinking)	Data transmission is happening on the link.
	OFF	LAN is not connected.
WAN	Solid Green	WAN link is established
	Green (Blinking)	Data transmission is happening on the link.
	OFF	WAN link is not connected.

Basic Router Configuration

This module provides basic configuration procedures for the Cisco 800M Series ISR and contains the following sections.

Configuring Global Parameters

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

SUMMARY STEPS

1. **configure terminal**
2. **hostname** *name*
3. **enable secret** *password*
4. **no ip domain-lookup**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`Router> enable`
	`Router# configure terminal`
		Enters global configuration mode, when using the console port.

Step 2	**hostname** *name* **Example:** `Router(config)# hostname Router`	Specifies the name for the router.
Step 3	**enable secret** *password* **Example:** `Router(config)# enable secret cr1ny5ho`	Specifies an encrypted password to prevent unauthorized access to the router.
Step 4	**no ip domain-lookup** **Example:** `Router(config)# no ip domain-lookup`	Disables the router from translating unfamiliar words (typos) into IP addresses.

Configuring Gigabit Ethernet WAN Interfaces

You can connect WAN interfaces either by using straight polarity connectors or reversed polarity connectors.

- **Straight Polarity**: If Mag-jack RJ45 connector has a dot or digit marked on front housing, it can be used with any type of cables.
- **Reversed Polarity**: If Mag-jack RJ45 connector has no dots or digit marked on front housing, it can be used with coupler and short cable (Cat5E UTP cable) to connect other devices which doesn't support auto polarity correction.

To configure Gigabit Ethernet (GE) WAN interfaces, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **configure terminal**
2. **interface gigabitethernet** *slot/port*
3. **ip address** *ip-address mask*
4. **no shutdown**
5. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**interface gigabitethernet** *slot/port* **Example:** `Router(config)# interface gigabitethernet 0/8`	Enters the configuration mode for a Gigabit Ethernet interface on the router. **Note** GigabitEthernet WAN Interfaces are 0/8 and 0/9 for Cisco C841M-8X ISR and 0/4 to 0/5 for Cisco C841M-4X
Step 3	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.12.2 255.255.255.0`	Sets the IP address and subnet mask for the specified GE interface.
Step 4	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the GE interface, changing its state from administratively down to administratively up.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode for the GE interface and returns to global configuration mode.

Configuring a Loopback Interface

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

To configure a loopback interface, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number*
3. **ip address** *ip-address mask*
4. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type number* **Example:** `Router(config)# interface Loopback 0`	Enters configuration mode for the loopback interface.
Step 3	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.108.1.1 255.255.255.0`	Sets the IP address and subnet mask for the loopback interface.
Step 4	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode for the loopback interface and returns to global configuration mode.

Example: Configuring the Loopback Interface

The loopback interface in this sample configuration is used to support Network Address Translation (NAT) on the virtual-template interface. This configuration example shows the loopback interface configured on the gigabit ethernet interface with an IP address of 200.200.100.1/24, which acts as a static IP address. The loopback interface points back to virtual-template1, which has a negotiated IP address.

```
!
interface loopback 0
ip address 200.200.100.1 255.255.255.0
ip nat outside
!
interface Virtual-Template1
ip unnumbered loopback0
no ip directed-broadcast
ip nat outside
!
```

Verifying the Loopback Interface Configuration

To verify that you have properly configured the loopback interface, enter the **show interface loopback** command as shown in the following example.

```
Router# show interface loopback 0
Loopback0 is up, line protocol is up
  Hardware is Loopback
  Internet address is 200.200.100.1/24
  MTU 1514 bytes, BW 8000000 Kbit, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, loopback not set
  Last input never, output never, output hang never
```

```
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/0, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
   0 packets input, 0 bytes, 0 no buffer
   Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   0 packets output, 0 bytes, 0 underruns
   0 output errors, 0 collisions, 0 interface resets
   0 output buffer failures, 0 output buffers swapped out
```

You can lso verify the loopback interface by using the **ping** command as shown in the following example.

```
Router# ping 200.200.100.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.200.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Configuring Command-Line Access

To configure parameters to control access to the router, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **line** [**aux** | **console** | **tty** | **vty**] *line-number*
3. **password** *password*
4. **login**
5. **exec-timeout** *minutes* [*seconds*]
6. **line** [**aux** | **console** | **tty** | **vty**] *line-number*
7. **password** *password*
8. **login**
9. **end**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**line** [**aux**	**console**	**tty**	**vty**] *line-number* **Example:** `Router(config)# line console 0`	Enters line configuration mode, and specifies the type of line.

	Command	Purpose
Step 3	**password** *password* **Example:** `Router(config)# password 5dr4Hepw3`	Specifies a unique password for the console terminal line.
Step 4	**login** **Example:** `Router(config-line)# login`	Enables password verification at the terminal login session.
Step 5	**exec-timeout** *minutes* [*seconds*] **Example:** `Router(config-line)# exec-timeout 5 30`	Sets the interval that the EXEC command interpreter waits until user input is detected. The default is 10 minutes. You can also optionally add seconds to the interval value.
Step 6	**line** [**aux** | **console** | **tty** | **vty**] *line-number* **Example:** `Router(config-line)# line vty 0 4`	Specifies a virtual terminal for remote console access.
Step 7	**password** *password* **Example:** `Router(config-line)# password aldf2ad1`	Specifies a unique password for the virtual terminal line.
Step 8	**login** **Example:** `Router(config-line)# login`	Enables password verifiation at the virtual terminal login session.
Step 9	**end** **Example:** `Router(config-line)# endRouter#`	Exits line configuration mode, and returns to privileged EXEC mode.

Configuring Gigabit Ethernet LAN Interfaces

To manually configure Gigabit Ethernet (GE) LAN interfaces, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **configure terminal**
2. **interface gigabitethernet** *slot/port*
3. **ip address** *ip-address mask*
4. **no shutdown**
5. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**interface gigabitethernet** *slot/port* **Example:** `Router(config)# interface gigabitethernet 0/1`	Enters the configuration mode for a Gigabit Ethernet interface on the router. **Note** GigabitEthernet LAN Interfaces are 0/0 to 0/7 for Cisco C841M-8X ISR and 0/0 to 0/3 for Cisco C841M-4X ISR.
Step 3	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.12.2 255.255.255.0`	Sets the IP address and subnet mask for the specified GE interface.
Step 4	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the GE interface, changing its state from administratively down to administratively up.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode for the GE interface and returns to global configuration mode.

Configuring Static Routes

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

To configure static routes, perform these steps in global configuration mode.

SUMMARY STEPS

1. **configure terminal**
2. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]}
3. **end**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} **Example:** `Router(config)# ip route 192.168.1.0 255.255.0.0 10.10.10.2`	Specifies the static route for the IP packets.
Step 3	**end** **Example:** `Router(config)# end`	Exits router configuration mode, and enters privileged EXEC mode.

Example: Configuring Static Routes

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

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

```
!
ip classless (default)
ip route 192.168.1.0 255.255.255.0 10.10.10.2!
```

Verifying Configuration

To verify that you have properly configured static routing, enter the **show ip route** command and look for static routes signified by the "S."

You should see verification output similar to the following:

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

Gateway of last resort is not set

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

Configuring Dynamic Routes

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

The Cisco routers can use IP routing protocols, such as Routing Information Protocol (RIP) or Enhanced Interior Gateway Routing Protocol (EIGRP), to learn routes dynamically. You can configure either of these routing protocols on your router.

- "Configuring Routing Information Protocol" section on page 13
- "Configuring Enhanced Interior Gateway Routing Protocol" section on page 15

Configuring Routing Information Protocol

To configure the RIP routing protocol on the router, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **configure terminal**
2. **router rip**
3. **version** {**1** | **2**}
4. **network** *ip-address*
5. **no auto-summary**
6. **end**

DETAILED STEPS

	Command	Task
Step 1	**configure terminal** **Example:** `Router> configure terminal`	Enters global configuration mode.
Step 2	**router rip** **Example:** `Router(config)# router rip`	Enters router configuration mode, and enables RIP on the router.
Step 3	**version** {**1** | **2**} **Example:** `Router(config-router)# version 2`	Specifies use of RIP version 1 or 2.
Step 4	**network** *ip-address* **Example:** `Router(config-router)# network 192.168.1.1`	Specifies a list of networks on which RIP is to be applied, using the address of the network of each directly connected network.
Step 5	**no auto-summary** **Example:** `Router(config-router)# no auto-summary`	Disables automatic summarization of subnet routes into network-level routes. This allows subprefix routing information to pass across classful network boundaries.
Step 6	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode, and enters privileged EXEC mode.

Example: RIP Configuration

The following configuration example shows RIP version 2 enabled in IP network 10.0.0.0 and 192.168.1.0.

To see this configuration, use the **show running-config** command from privileged EXEC mode.

```
Router# show running-config
router rip
 version 2
 network 10.0.0.0
 network 192.168.1.0
 no auto-summary
!
```

Verifying RIP Configuration

To verify that you have properly configured RIP, enter the **show ip route** command and look for RIP routes signified by "R" as shown in this example.

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

Gateway of last resort is not set

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

Configuring Enhanced Interior Gateway Routing Protocol

To configure Enhanced Interior Gateway Routing Protocol (EGRP), perform these steps.

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp** *as-number*
3. **network** *ip-address*
4. **end**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** `Router> configure terminal`	Enters global configuration mode.
Step 2	**router eigrp** *as-number* **Example:** `Router(config)# router eigrp 109`	Enters router configuration mode, and enables EIGRP on the router. The autonomous-system number identifies the route to other EIGRP routers and is used to tag the EIGRP information.
Step 3	**network** *ip-address* **Example:** `Router(config)# network 192.145.1.0`	Specifies a list of networks on which EIGRP is to be applied, using the IP address of the network of directly connected networks.

	Command	Purpose

Step 4	**end**
	
	Example:
	`Router(config-router)# end`
	`Router#`
		Exits router configuration mode, and enters privileged EXEC mode.

Example: Configuring EIGRP

This configuration example shows the EIGRP routing protocol enabled in IP networks 192.145.1.0 and 10.10.12.115. The EIGRP autonomous system number is 109.

To see this configuration use the **show running-config** command, beginning in privileged EXEC mode.

```
Router# show running-config...
!
router eigrp 109
    network 192.145.1.0
        network 10.10.12.115
!
...
```

Verifying EIGRP Configuration

To verify that you have properly configured EIGRP, enter the **show ip route** command, and look for EIGRP routes indicated by "D " as shown in the following example:

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

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.108.1.0 is directly connected, Loopback0
D    3.0.0.0/8 [90/409600] via 2.2.2.1, 00:00:02, Ethernet0/0
```

Configuring Image and Configuration Recovery Using the Push Button

A push or reset button is available on the rear side of the Cisco 800M Series ISR and it is designed to provide a disaster recovery method for the router.

Push button can be useful for recovery during one of the two scenarios:

- During ROMMON initialization
- For loading a specific configuration file without accessing the router IOS prompt after IOS is up and running.

Push Button Behavior During ROMMON Initialization

Table 2-1 shows the high level functionality when the push button is pressed during ROMMON initialization.

Table 2-1 Push Button Functionality During ROMMON Initialization

ROMMON Behavior	IOS Behavior
• Boots using default baud rate. • Performs auto-boot. • Loads the *.default image if available on compact flash	If the configuration named *.cfg is available in NVRAM storage or flash storage, IOS will perform a backup of the original configuration and boots up using this configuration.

Push Button Behavior When IOS is up and Running

If you press the push button for more than three seconds and then release the push button after IOS is up and running, IOS detects this event and looks for configuration files in the order of priority.If the IOS finds the configuration file, it copies the configuration file to the startup configuration file. Then the router reloads itself and the new configuration takes effect. If the configuration files cannot be found, pressing reset button has no effect.

The order of priority in which the router looks for configuration file is given as follows:

1. usbflash0:customer-config.*SN*
2. usbflash0:customer-config
3. flash:customer-config.*SN*
4. flash:customer-config

Note *SN* is the hardware serial number.

Configuring 800M Series ISR using Zero Touch Deployment

The Zero Touch Deployment (ZTD) through USB feature in Cisco 800M Series ISRs is an ease-of-use feature that loads a customized configuration from a USB flash drive. This feature requires that the router has no startup configuration in its nonvolatile RAM (NVRAM). The feature also requires that a valid configuration file, with the filename extension **.cfg**, is stored in the USB flash drive. A valid configuration file can be created by saving the running configuration of a router to flash, USB flash, or to a TFTP Server.

When a router with no startup configuration boots up, it checks for a valid configuration file within the USB flash drive. The pre-requisites for deployment using the Zero Touch Deployment through USB feature are:

- Boot up router with no startup-configuration.
- Cisco USB flash drive inserted in the first available USB slot.

- A valid configuration file in ASCII text with the filename extension **.cfg**

If the USB flash drive has multiple .cfg files, the router chooses the one with the highest index number in the USB Flash drive. To avoid loading an incorrect .cfg file, ensure that there is only one .cfg file in the USB flash drive.

The Cisco 800M Series ISR uses second core and it is actively used in detecting USB flash drive if 3G Wireless WAN module is present on the router. If 3G Wireless WAN module is not present, USB flash drive is detected by the IOS. When 3G Wireless WAN module is present, USB detection is a bit delayed for the Cisco 800M series ISR due to the delay in second core initialization. While system startup is in progress and push button is pressed, a timer is started to check the completion of second core initialization. For some reason if second core takes more time, system reports an error message and continues the normal start up. After second core initialization router waits up to 10 seconds for USB detection and then complete the configuration. In case the USB flash drive does not contain a deployment configuration, router enters the configuration mode.

Configuring 3G Wireless WAN

This chapter provides information about configuring the 3G Wireless WAN interface on Cisco 800M Series ISRs and contains the following sections:

Overview of 3G Wireless WAN

3G Wireless WAN offers a highly secure, simplified, and cost-effective WAN alternative to DSL or Frame Relay. In areas where terrestrial broadband services (cable, DSL, or T1) are not available or are expensive, 3G Wireless WAN connectivity can be a viable alternative. Using the integrated services available on the Cisco 800M Series ISR, 3G Wireless WAN can provide instant and mobile communications during disasters and service outages. Cisco 800M Series ISRs support GSM and CDMA 3G Wireless WAN networks through the pluggable 3G WAN module. The primary application for 3G Wireless WAN module is WAN connectivity as a backup data link for critical data applications. However, the 3G wireless interface can also function as the primary WAN connection for the router.

3G Wireless WAN Features Supported on Cisco 800M Series ISR

3G Wireless WAN module on the Cisco 800M Series ISR is based on Sierra Wireless 9090 modem that supports both GSM and CDMA. Technology mode is auto selected based on the current active firmware.

The following table lists the GSM/CDMA modes and the supported frequency bands.

Table 3-1 Supported Cellular Modes and Frequencies

Mode	Frequency
GSM/GPRS/EDGE	850/900/1800/1900 MHz
WCDMA	800/850/1900/2100 MHz
CDMA (EVDO Rev A/ 1xRTT)	800/1900 MHz

Cisco 800M Series ISRs support the following 3G Wireless WAN features:

- Dual SIM
- SIM lock and unlock capabilities
- Multiple Profile
- Crash Dump Support
- Diagnostic Monitor Logging
- Firmware upgrade
- Entity MIB

Table 3-2 describes the LEDs on the 3G Wireless WAN module.

Table 3-2 LEDs on the 3G Wireless WAN Module

LED	LED Color	Description
3G RSSI	Solid green	High RSSI (–60 dBm or higher)
	3 Blinks and long pause	Medium RSSI (–74 to –60 dBm)
	2 Blinks and long pause	-89 to -75 dBm
	1 Blink and long pause	-109 to -90dBm
	Off	Low RSSI (less than –100 dBm)
SIM 0	Solid green	SIM 0 is active and connected to UMTS/EVDO.
	Green (1 blink)	SIM 0 is active and connected to GSM/1xRTT.
SIM 1	Solid green	SIM 1 is active and connected to UMTS/EVDO.
	Green (1 blink)	SIM 1 is active and connected to GSM/1xRTT.
WWAN	Green (fast blinking)	Traffic is flowing through the WAN link.

Pre-requisites for Configuring 3G Wireless WAN on Cisco 800M Series ISRs

The following are prerequisites to configuring the 3G Wireless WAN interface:

- You must have wireless service from a carrier, and you must have network coverage where your router will be physically placed. For a complete list of supported carriers, see the product data sheet.
- You must subscribe to a service plan with a wireless service provider and obtain a Subscriber Identity Module (SIM) card from the service provider. For CDMA, you should get an active Removable User Identity Module (RUIM) card.
- You must check your LEDs for signal strength, as described in Table 3-2.
- To configure your GSM data profile, you need the following information from your service provider:
  - Username
  - Password
  - Access point name (APN)

Restrictions for Configuring 3G Wireless WAN on the Cisco 800M Series ISR

The following restrictions apply to configuring the Cisco 3G wireless interface:

- A data connection can be originated only by the 3G wireless interface. Remote dial-in is not supported.
- Because of the shared nature of wireless communications, the experienced throughput varies depending on the number of active users or the amount of congestion in a given network.
- Cellular networks have higher latency than wired networks. Latency rates depend on the technology and carrier. Latency may be higher when there is network congestion.
- Any restrictions that are part of the terms of service from your carrier also apply to the Cisco 3G wireless interface.
- Short Message Service (SMS) is not supported.
- Global Positioning System (GPS) is not supported.
- Mobile Equipment Personalization (MEP) is not supported.
- Public Land Mobile Network (PLMN) search is not supported.

Note Only one 3G module is supported at a time on the Cisco 800M Series Router. If two 3G modules are present in the 800M Series Router, the 3G module in the second slot will be powered down.

Configuring GSM Mode on Cisco 800M Series ISRs

To configure GSM mode on the 3G cellular Wireless WAN interface, perform these procedures:

Data Account Provisioning

Note To provision your modem, you must have an active wireless account with a service provider. A SIM card must be installed in the GSM 3G wireless module.

To provision your data account, follow these procedures:

Verifying Signal Strength and Service Availability

To verify the signal strength and service availability on your modem, use the following commands in privileged EXEC mode.

Table 3-3 Commands for Verifying Signal Strength

Command or Action	Purpose
show cellular *unit* **network**	Displays information about the carrier network, cell site, and available service.
show cellular *unit* **hardware**	Displays the cellular modem hardware information.
show cellular *unit* **connection**	Displays the current active connection state and data statistics.
show cellular *unit* **radio**	Shows the radio signal strength.
show cellular *unit* **profile**	Displays information about the modem data profiles created.
show cellular *unit* **security**	Shows the security information for the modem, such as active SIM and modem lock status.
show cellular *unit* **all**	Shows consolidated information about the modem. The profiles that were created, the radio signal strength, the network security, and so on.

Note In the configuration procedures given in this chapter, the *unit* argument identifies the router slot, WIC slot, and port separated by slashes (0/0/0).

Configuring a GSM Modem Data Profile

Enter the following command to configure or create a new modem data profile in privileged EXEC mode.

Table 3-4 Configuring a GSM Modem Data Profile

Command	Purpose
cellular *unit* **profile create** *profile-number apn authentication username password* **Example:** `Router# cellular 0/0/0 profile create 3 apn.com chap GSM GSMPassword`	Configures a new modem data profile. • *profile-number*—Specifies a number for the profile that you are creating. You can create up to 16 profiles. **Note** For GSM, default data profile is profile1. • *apn*—Specifies the access point name. You must get this information from your service provider. • *authentication*—Specifies the type of authentication, for example, CHAP, PAP. • *username*—Specifies the user name provided by your service provider. • *password*—Specifies the password provided by your service provider.

Note For deleting a GSM data profile, use the **cellular** *unit* **profile delete** *profile-number* command.

Example: Configuring GSM Data Profile

This example shows the GSM profiles created on the cellular interface 0/1/0.

```
Router# show cellular 0/1/0 profile

Profile 1 = ACTIVE*
--------
PDP Type = IPv4
PDP address = 117.96.4.183
Access Point Name (APN) = airtelgprs.com
Authentication = None
Username:
Password:
Primary DNS address = 125.22.47.102
Secondary DNS address = 125.22.47.103

Profile 4 = INACTIVE
--------
PDP Type = IPv4
Access Point Name (APN) = aircel.com
Authentication = CHAP
Username: aircell
Password: aircel


Profile 11 = INACTIVE
--------
PDP Type = IPv4
Access Point Name (APN) = vodafone
Authentication = None
Username:
```

```
Password:


Profile 15 = INACTIVE
--------
PDP Type = IPv4
Access Point Name (APN) = aircel.com
Authentication = CHAP
Username: aircell
Password: aircel

* - Default profile

Configured default profile for active SIM 0 is profile 1.
```

Setting up a Data Call

A data call is a call setup through a signaling protocol on the Public Switching Telephony Network (PSTN) to a Network Access Server (NAS) to transfer data, either as a byte stream (for example, terminal emulation) or in a packet format (for example, PPP packets) from a data terminal (such as a PC) to a data network.

To setup a data call, perform the following tasks:

- Configuring a Cellular Interface, page 24
- Configuring DDR, page 25
- Configuring DDR Backup, page 27

Configuring a Cellular Interface

To configure the cellular interface, enter the following commands, beginning in privileged EXEC mode.

SUMMARY STEPS

1. **configure terminal**
2. **interface cellular** *unit*
3. **encapsulation slip**
4. **asynchronous mode interactive**
5. **ip address negotiated**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode from the terminal.
Step 2	**interface cellular** *unit* **Example:** `Router(config)# interface cellular 0/0/0`	Specifies the cellular interface.
Step 3	**encapsulation slip** **Example:** `Router(config-if)# encapsulation slip`	Specifies slip encapsulation for an interface configured for dedicated asynchronous mode or dial-on-demand routing.
Step 4	**asynchronous mode interactive** **Example:** `Router(config-if)# asynchronous mode interactive`	Returns a line from dedicated asynchronous network mode to interactive mode, enabling the **slip** and **ppp** commands in privileged EXEC mode.
Step 5	**ip address negotiated** **Example:** `Router(config-if)# ip address negotiated`	Specifies that the IP address for a particular interface is dynamically obtained.

Note When the cellular interface requires a static IP address, the address may be configured as **ip address negotiated**. Through IP Control Protocol (IPCP), the network ensures that the correct static IP address is allocated to the device. If a tunnel interface is configured with the **ip address unnumbered** *cellular interface* command, the actual static IP address must be configured under the cellular interface, in place of **ip address negotiated**. For a sample cellular interface configuration, see the "Basic Cellular Interface Configuration" section on page 3-32.

Configuring DDR

Perform these steps to configure dial-on-demand routing (DDR) for the cellular interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface cellular** *unit*
3. **dialer in-band**
4. **dialer idle-timeout** *seconds*
5. **dialer string** *string*
6. **dialer group** *number*

7. **exit**
8. **dialer-list** *dialer-group* **protocol** *protocol-name* {**permit** | **deny** | **list** *access-list-number* | **access-group**}
9. **ip access-list** *access-list-number* **permit** *ip-source-address*
10. **line** *unit*
11. **script dialer** *regexp*
12. **exit**
13. **chat-script** *script-name* **"" "AT!CALL** *profile-number#***" TIMEOUT** *timeout-value* **"OK"**
14. **interface cellular** *unit*
15. **dialer string** *string*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**interface cellular** *unit* **Example:** `Router(config)# interface cellular 0/0/0`	Specifies the cellular interface.
Step 3	**dialer in-band** **Example:** `Router(config-if)# dialer in-band`	Enables DDR and configures the specified serial interface for in-band dialing.
Step 4	**dialer idle-timeout** *seconds* **Example:** `Router(config-if)# dialer idle-timeout 30`	Specifies the duration of idle time, in seconds, after which a line will be disconnected.
Step 5	**dialer string** *string* **Example:** `Router(config-if)# dialer string multimode`	Specifies the number or string to dial. Use the name of the chat script here.
Step 6	**dialer-group** *number* **Example:** `Router(config-if)# dialer-group 1`	Specifies the number of the dialer access group to which a specific interface belongs.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config-if)# exit`	Enters the global configuration mode.
Step 8	**dialer-list** *dialer-group* **protocol** *protocol-name* {**permit** | **deny** | **list** *access-list-number* | **access-group**} **Example:** `Router(config)# dialer-list 1 protocol ip list 1`	Creates a dialer list for traffic of interest and permits access to an entire protocol.
Step 9	**ip access-list** *access-list-number* **permit** *ip-source-address* **Example:** `Router(config)# ip access list 1 permit any`	Defines traffic of interest.
Step 10	**line** *unit* **Example:** `Router(config-line)# line 3`	Specifies the line configuration mode.
Step 11	**script dialer** *regexp* **Example:** `Router(config-line)# script-dialer multimode`	Specifies a default modem chat script.
Step 12	**exit** **Example:** `Router (config-line)# exit`	Exits line configuration mode.
Step 13	**chat-script** *script-name* **"" "AT!CALL" TIMEOUT** *timeout-value* **"OK"** **Example:** `Router(config)# chat-script multimode "" "AT!CALL" TIMEOUT 60 "OK"`	Defines the Attention Dial Tone (ATDT) commands when the dialer is initiated.
Step 14	**interface cellular** *unit* **Example:** `Router(config)# interface cellular 0`	Specifies the cellular interface.
Step 15	**dialer string** *string* **Example:** `Router(config)# dialer string multimode`	Specifies the dialer script (defined using the **chat script** command).

Configuring DDR Backup

To monitor the primary connection and initiate the backup connection when needed, the router can use the following method:

- Floating Static Route—The route through the backup interface has an administrative distance that is greater than the administrative distance of the primary connection route and therefore would not be in the routing table until the primary interface goes down.

Configuring DDR Backup Using Floating Static Route

To configure a floating static default route on the secondary interface beginning in the global configuration mode, perform the following tasks.

Note Make sure you have ip classless enabled on your router.

SUMMARY STEPS

1. **configure terminal**
2. **ip route** *network-number network-mask* {*ip-address* | *interface*} [*administrative distance*] [**name** *name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**ip route** *network-number network-mask* {*ip-address* | *interface*} [*administrative distance*] [**name** *name*] **Example:** `Device(config-if)# ip route 209.165.200.225 255.255.255.224 Dialer 2 253 name name1`	Establishes a floating static route with the configured administrative distance through the specified interface. **Note** A higher administrative distance should be configured for the route through the backup interface so that it is used only when the primary interface is down.

Configuring CDMA Mode on Cisco 800M Series ISRs

Perform the following procedures for configuring CDMA mode on Cisco 800M Series 3G WWAN module:

- Activating the Modem, page 29
- Setting up a Data Call, page 29

Activating the Modem

Manual activation of the CDMA modem is not supported. The activation and provisioning procedures may differ depending upon your carrier. To activate the CDMA modem, contact your service provider.

Setting up a Data Call

Perform these procedures to set up a data call for CDMA mode.

- Configuring a Cellular Interface, page 24
- Configuring DDR, page 25
- Configuring DDR Backup, page 27

Configuring a Cellular Interface

To configure the cellular interface, enter the following commands, beginning in privileged EXEC mode.

SUMMARY STEPS

1. **configure terminal**
2. **interface cellular** *unit*
3. **encapsulation slip**
4. **asynchronous mode interactive**
5. **ip address negotiated**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** Example: `Router# configure terminal`	Enters global configuration mode from the terminal.
Step 2	**interface cellular** *unit* Example: `Router(config)# interface cellular 0/0/0`	Specifies the cellular interface.
Step 3	**encapsulation slip** Example: `Router(config-if)# encapsulation slip`	Specifies slip encapsulation for an interface configured for dedicated asynchronous mode or dial-on-demand routing.

	Command or Action	Purpose
Step 4	**asynchronous mode interactive** **Example:** `Router(config-if)# asynchronous mode interactive`	Returns a line from dedicated asynchronous network mode to interactive mode, enabling the **slip** and **ppp** commands in privileged EXEC mode.
Step 5	**ip address negotiated** **Example:** `Router(config-if)# ip address negotiated`	Specifies that the IP address for a particular interface is obtained via PPP and IPCP address negotiation.

Configuring DDR

Perform these steps to configure dial-on-demand routing (DDR) for the cellular interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface cellular** *unit*
3. **dialer in-band**
4. **dialer idle-timeout** *seconds*
5. **dialer string** *string*
6. **dialer group** *number*
7. **exit**
8. **dialer-list** *dialer-group* **protocol** *protocol-name* {**permit** | **deny** | **list** *access-list-number* | **access-group**}
9. **ip access-list** *access-list-number* **permit** *ip-source-address*
10. **line** *unit*
11. **script dialer** *regexp*
12. **exit**
13. **chat-script** *script name* **"" "AT!CALL** *profile-number#***" TIMEOUT** *timeout-value* **"OK"**
14. **interface cellular** *unit*
15. **dialer string** *string*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**interface cellular** *unit* **Example:** `Router(config)# interface cellular 0/0/0`	Specifies the cellular interface.
Step 3	**dialer in-band** **Example:** `Router(config-if)# dialer in-band`	Enables DDR and configures the specified serial interface for in-band dialing.
Step 4	**dialer idle-timeout** *seconds* **Example:** `Router(config-if)# dialer idle-timeout 30`	Specifies the duration of idle time, in seconds, after which a line will be disconnected.
Step 5	**dialer string** *string* **Example:** `Router(config-if)# dialer string multimode`	Specifies the number or string to dial. Use the name of the chat script here.
Step 6	**dialer-group** *number* **Example:** `Router(config-if)# dialer-group 1`	Specifies the number of the dialer access group to which a specific interface belongs.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Enters the global configuration mode.
Step 8	**dialer-list** *dialer-group* **protocol** *protocol-name* {**permit** | **deny** | **list** *access-list-number* | **access-group**} **Example:** `Router(config)# dialer-list 1 protocol ip list 1`	Creates a dialer list for traffic of interest and permits access to an entire protocol.
Step 9	**ip access-list** *access-list-number* **permit** *ip-source-address* **Example:** `Router(config)# ip access-list 1 permit any`	Defines traffic of interest.

	Command or Action	Purpose
Step 10	**line** *unit* **Example:** `Router(config-line)# line 0/0/0`	Specifies the line configuration mode.
Step 11	**script dialer** *regexp* **Example:** `Router(config-line)# script-dialer multimode`	Specifies a default modem chat script.
Step 12	**exit** **Example:** `Router (config-line)# exit`	Exits line configuration mode.
Step 13	**chat-script** *script-name* **"" "AT!CALL" TIMEOUT** *timeout-value* **"OK"** **Example:** `Router(config)# chat-script multimode "" "AT!CALL" TIMEOUT 60 "OK"`	Defines the Attention Dial Tone (ATDT) commands when the dialer is initiated.
Step 14	**interface cellular** *unit* **Example:** `Router(config)# interface cellular 0/0/0`	Specifies the cellular interface.
Step 15	**dialer string** *string* **Example:** `Router(config)# dialer string multimode`	Specifies the dialer script (defined using the **chat script** command).

Configuring DDR Backup

The configuration tasks for configuring DDR backup is same for GSM and CDMA. To configure DDR back up for CDMA, See the Configuring DDR Backup, page 27 for GSM and perform the steps.

Configuration Examples

This section provides the following configuration examples:

Basic Cellular Interface Configuration

The following example shows how to configure a cellular interface (GSM/CDMA) to be used as a primary WAN connection. It is configured as the default route.

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

```
!
chat-script multimode "" "AT!CALL1" TIMEOUT 20 "OK"
 interface Cellular0/0/0
 ip address negotiated
 encapsulation slip
 load-interval 30
 dialer in-band
 dialer idle-timeout 0
 dialer string multimode
 dialer-group 1
 no peer default ip address
 async mode interactive
 routing dynamic
 ip route 0.0.0.0 0.0.0.0 Cellular0/0/0
 dialer-list 1 protocol ip permit
 line 3
 script dialer multimode
 modem InOut
 no exec
 transport input all
 transport output all
```

Tunnel over Cellular Interface Configuration

The following example shows how to configure the static IP address when a tunnel interface is configured with the **ip address unnumbered** *cellular interface* command:

```
interface Tunnel2
 ip unnumbered Cellular0/0/0
 tunnel source Cellular0/0/0
 tunnel destination 128.107.248.254

interface Cellular0/0/0
 bandwidth receive 1400000
 ip address 23.23.0.1 255.255.0.0
 ip nat outside
 ip virtual-reassembly
 encapsulation slip
 no ip mroute-cache
 dialer in-band
 dialer idle-timeout 0
 dialer string dial<carrier>
 dialer-group 1
 async mode interactive
 ! traffic of interest through the tunnel/cellular interface
ip route 10.10.0.0 255.255.0.0 Tunnel2
```

Configuring Dual SIM for Cellular Networks

The Dual SIM feature implements auto-switch and fail over between two cellular networks. This feature is enabled by default with SIM slot 0 being the primary slot and slot 1 being the secondary (fail over) slot.

Usage Guidelines for Configuring a Dual SIM

Follow these guidelines while you configure a dual SIM:

- Configure the SIM profile for slots 0 and 1 using the **sim profile** command.
- For auto-switch and fail over to work, configure the chat script without a specific profile number.
- If SIM profile is not configured, profile #1 is used by default.
- If fail over timer is not configured, the default failover timeout is 2 minutes.
- If SIM primary slot is not configured, the default primary SIM is slot 0.

Note Dual SIM feature is supported only when the same firmware image is used for both the SIM cards.

SUMMARY STEPS

1. **configure terminal**
2. **controller cellular** *unit*
3. **sim primary** *slot*
4. **sim max-retry** *number*
5. **sim authenticate** [**0** | **7**] *pin* **slot** {**0** | **1**}
6. **failover** *timeout-period*
7. **sim profile** *number* [**ims** *number*] **slot** {**0** | **1**}

DETAILED STEPS

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

Step 2	**controller cellular** *unit*
	
	Example:
	`Device(config)# controller cellular 0/0`
	
	or
	`Device(config)# controller cellular 0/1`
		Enters the cellular controller configuration mode.

Step 3	**sim primary** *slot* **Example:** `Device(config-controller)# sim primary slot 1`	(Optional) Enters either slot number 0 or 1 of the primary SIM.
Step 4	**sim max-retry** *number* **Example:** `Device(config-controller)# gsm sim max-retry 20`	(Optional) Specifies the maximum number of fail over retries from 1 to 65535. The default value is 10.

	Command or Action	Purpose
Step 5	**sim authenticate [0 | 7]** *pin* **slot {0 | 1}** **Example:** `Device(config-controller)# gsm sim authenticate 0 1234 slot 0`	Authenticates the SIM CHV1 code
Step 6	**failovertimer** *timeout-period* **Example:** `Device(config-controller)# failovertimer 6`	(Optional) By default, the fail over time period is 2 minutes before the primary SIM switches over to the secondary SIM if service becomes unavailable. Specify a fail over timeout value between 1 and 7 minutes before a switchover occurs.
Step 7	**sim profile** *number* **slot** {**0** | **1**} **Example:** `Device(config-controller)# sim profile 1 slot 0`	Applies the configured profile number to the SIM and its slot number. The default (primary) slot is 0. You must also identify the primary and secondary SIM for the configured profile when two SIMs are presented.

Note Before you start the modem crash dump, turn off the SIM switch over by configuring the **sim max-retry** 0 command.

Configuration Examples

The following example shows how to configure a dual SIM:

```
router# configure terminal
router(config)# controller Cellular 0/0
router(config-controller)# sim profile 1 slot 0
router(config-controller)# sim primary slot 1
router(config-controller)# sim max-retry 20
router(config-controller)# sim failovertimer 5
```

Confguring SIM Lock and Unlock

Use the following commands for locking or unlocking the SIM.

.

Table 3-5 Commands for Manually Switching the SIM

Command	Purpose
cellular sim {lock | unlock}	Locks or unlocks the SIM.
cellular *unit* **sim [lock | unlock]** *pin*	Locks or unlocks the SIM.
cellular *unit* **sim unlock** *newpin*	Unlocks the SIM.

Upgrading Modem Firmware

The 3G Wireless WAN module for Cisco 800M Series ISRs comes with SL9090 modem from Sierra Wireless. The firmware for the modem is upgradable using Cisco IOS commands. The firmware can be downloaded from the wireless software download page on Cisco.com.

Use the following procedure to upgrade the modem firmware:

Note Before upgrading the modem to a new firmware version, please check if the new firmware version has been certified by your wireless service provider. Using an uncertified firmware version on the modem may impact the wireless service provider network adversely. See the following web link for the latest certified firmware version for your carrier and IOS compatibility: http://www.cisco.com/en/US/prod/routers/networking_solutions_products_genericcontent0900aecd80601f7e.html

Step 1 Go to the 3G firmware download website and select the carrier:

http://software.cisco.com/download/navigator.html?mdfid=279119319&flowid=6999

Step 2 Download the appropriate firmware release under Wireless Integrated Switches and Routers.

Step 3 Copy the files to the device's flash.

Step 4 Use the following command to initiate the firmware upgrade process:

microcode reload cellular *bay slot slot* **modem-provision** *flash:*

Switching Modem Firmware Image

The 3G Wireless WAN module can support firmware images for GSM and CDMA and support carrier switching. Only one firmware image is supported at a time. Auto switching between different firmware packages is not supported.

You can use the following commands for switching modem firmware:

Command	Description
show cellular *unit* **microcode**	Displays the list firmware images available on the modem.
cellular 0/0/0 microcode activate *firmware-id*	Activates the specified modem firmware.

Note Once you perform the modem firmware switching, you need to perform the modem power cycle using **test cellular** *unit* **modem-power-cycle** command. To enable test commands, you should enter the **service internal** command in global configuration mode.

This example shows displaying the list of firmware images using **show cellular** *unit* **microcode** command and activating a specific firmware package using **cellular microcode activate** *firmware-id* command.

```
Router# show cellular 0/1/0 microcode
Modem:
-------------
 ID   Carrier               Technology  Version     Status
 1    Verizon               CDMA        02000007    INACTIVE
 2    Generic               UMTS        02010303    ACTIVE
 3    Sprint                   CDMA        02010001    INACTIVE
 4    China Telecom  CDMA        02000001    INACTIVE

Router# cellular 0/0/0 microcode activate 2
**************************************************
 The interface will be Shut Down for Firmware Activation  This will terminate any active
data connections.
**************************************************
Please wait while selected firmware is activated ...
Modem radio has been turned off.
*Feb  6 13:08:13.627: %CISCO800-2-MODEM_DOWN: Cellular0/0/0 modem is now DOWN........
Firmware activated successfully
```

Related Documents

Topic	Document Title
GSM	Configuring Cisco EHWIC and 880G for 3.7G (HSPA+)/3.5G (HSPA)
CDMA	Configuring Cisco EHWIC and 880G for 3G (EV-DO Rev A)
DM Log Collection and modem crashdump support	Cisco 3G and 4G Serviceability Enhancement User Guide
MIB • CISCO-ENTITY-VENDORTYPE-OID-MIB • CISCO-WAN-3G-MIB	MIB Locator Tool

Configuring the Serial Interface

This chapter describes configuring the serial interface for Cisco 800M Series ISRs in the following sections:

Configuring the Serial Interface

The Cisco 800M Series Integrated Services Router (ISR) provides serial WAN connectivity to remote sites using Cisco High-Level Data Link Control (HDLC), Point-to-Point Protocol (PPP), or Frame Relay encapsulation through the pluggable, serial WAN interface module. The Cisco 800M Series ISR supports both synchronous and asynchronous modes of communication.

Features Supported by Serial Module

The Cisco 800M Series ISR has 2 WAN slots that can host single-port serial module or multi-mode 3G module and supports the following combinations.

- Serial module in slot 0 and 3G module in slot 1
- 3G module in slot 0 and serial module in slot 1
- Serial module in slot 0 and serial module in slot 1

The features supported by the single-port serial module on Cisco 800M Series ISR is given as follows:

- Supports the following encapsulations :
  - HDLC
  - PPP
  - Frame Relay
  - Serial Line Internet Protocol (SLIP)
- Supports the following serial protocols

  - EIA-232
  - EIA-449
  - EIA-530
  - EIA-530A
  - V.35
  - X.21
- Supports synchronous speed of up to 8 Mbps
- Supports asynchronous speed of up to 115.2 kbps
- Supports network clock synchronization

Cisco 800M Series ISRs use Cisco smart serial connectors. Information about the cables supported by Cisco 800M Series ISRs are provided in Table 4-1.

Table 4-1 Smart Serial Cabling for Cisco 800M Series ISRs

Product Number	Cable Type	Length	Connector Type
CAB-SS-V35MT	V.35 DTE	10 ft (3m)	Male
CAB-SS-V35FC	V.35 DCE	10 ft (3m)	Female
CAB-SS-232MT	EIA/TIA-232 DTE	10 ft (3m)	Male
CAB-SS-232FC	EIA/TIA-232 DTE	10 ft (3m)	Female
CAB-SS-449MT	EIA/TIA-449 DTE	10 ft (3m)	Male
CAB-SS-449FC	EIA/TIA-449 DTE	10 ft (3m)	Female
CAB-SS-X21MT	X.21 DTE	10 ft (3m)	Male
CAB-SS-X21FC	X.21 DTE	10 ft (3m)	Female
CAB-SS-530MT	EIA/TIA-530 DTE	10 ft (3m)	Male
CAB-SS-530AMT	EIA/TIA-232 DTE	10 ft (3m)	Male

Table 4-2 describes the LEDs on the Cisco 800M series serial WAN module.

Table 4-2 LEDs on the Serial WAN Module

LED Name	Color/ Status	Description
CONN	Green	Indicates the interface status and shows that line protocol is up.
	OFF	Shows that the line protocol is down.
LOOP BACK	Green	Indicates that the hardware loopback status is configured on the serial interface.
	OFF	Indicates that loopback is not configured.

Information About Configuring Serial Interfaces

To configure serial interfaces, you should understand the following concept:

Cisco HDLC Encapsulation

Cisco High-Level Data Link Controller (HDLC) is the Cisco proprietary protocol for sending data over synchronous serial links. Cisco HDLC also provides a simple control protocol called Serial Line Address Resolution Protocol (SLARP) to maintain serial link keepalives. Cisco HDLC is the default for data encapsulation at Layer 2 (data link) of the Open System Interconnection (OSI) stack for efficient packet delineation and error control.

Note Cisco HDLC is the default encapsulation type for the serial interfaces.

When the encapsulation on a serial interface is changed from HDLC to any other encapsulation type, the configured serial subinterfaces on the main interface inherit the newly changed encapsulation and they do not get deleted.

Cisco HDLC uses keepalives to monitor the link state, as described in the "Keepalive Timer" section on page 4-43.

PPP Encapsulation

PPP is a standard protocol used to send data over synchronous serial links. PPP also provides a Link Control Protocol (LCP) for negotiating properties of the link. LCP uses echo requests and responses to monitor the continuing availability of the link.

Note When an interface is configured with PPP encapsulation, a link is declared down and full LCP negotiation is re-initiated after five echo request (ECHOREQ) packets are sent without receiving an echo response (ECHOREP).

PPP provides the following Network Control Protocols (NCPs) for negotiating properties of data protocols that will run on the link:

- IP Control Protocol (IPCP) to negotiate IP properties
- Multiprotocol Label Switching control processor (MPLSCP) to negotiate MPLS properties
- Cisco Discovery Protocol control processor (CDPCP) to negotiate CDP properties
- IPv6CP to negotiate IP Version 6 (IPv6) properties
- Open Systems Interconnection control processor (OSICP) to negotiate OSI properties

PPP uses keepalives to monitor the link state, as described in the "Keepalive Timer" section on page 4-43.

PPP supports the following authentication protocols, which require a remote device to prove its identity before allowing data traffic to flow over a connection:

- Challenge Handshake Authentication Protocol (CHAP)—CHAP authentication sends a challenge message to the remote device. The remote device encrypts the challenge value with a shared secret and returns the encrypted value and its name to the local router in a response message. The local router attempts to match the remote device's name with an associated secret stored in the local username or remote security server database; it uses the stored secret to encrypt the original challenge and verify that the encrypted values match.
- Microsoft Challenge Handshake Authentication Protocol (MS-CHAP)—MS-CHAP is the Microsoft version of CHAP. Like the standard version of CHAP, MS-CHAP is used for PPP authentication; in this case, authentication occurs between a personal computer using Microsoft Windows and a Cisco router or access server acting as a network access server.
- Password Authentication Protocol (PAP)—PAP authentication requires the remote device to send a name and a password, which are checked against a matching entry in the local username database or in the remote security server database.

Use the **ppp authentication** command in interface configuration mode to enable CHAP, MS-CHAP, and PAP on a serial interface.

Note Enabling or disabling PPP authentication does not effect the local router's willingness to authenticate itself to the remote device.

Multilink PPP

Multilink Point-to-Point Protocol (MLPPP) is supported on the Cisco 800M Series ISR serial interface. MLPPP provides a method for combining multiple physical links into one logical link. The implementation of MLPPP combines multiple PPP serial interfaces into one multilink interface. MLPPP performs the fragmenting, reassembling, and sequencing of datagrams across multiple PPP links.

MLPPP provides the same features that are supported on PPP Serial interfaces with the exception of QoS. It also provides the following additional features:

- Fragment sizes of 128, 256, and 512 bytes
- Long sequence numbers (24-bit)
- Lost fragment detection timeout period of 80 ms
- Minimum-active-links configuration option
- LCP echo request/reply support over multilink interface
- Full T1 and E1 framed and unframed links

Keepalive Timer

Cisco keepalives are useful for monitoring the link state. Periodic keepalives are sent to and received from the peer at a frequency determined by the value of the keepalive timer. If an acceptable keepalive response is not received from the peer, the link makes the transition to the down state. As soon as an acceptable keepalive response is obtained from the peer or if keepalives are disabled, the link makes the transition to the up state.

Note The **keepalive** command applies to serial interfaces using HDLC or PPP encapsulation. It does not apply to serial interfaces using Frame Relay encapsulation.

For each encapsulation type, a certain number of keepalives ignored by a peer triggers the serial interface to transition to the down state. For HDLC encapsulation, three ignored keepalives causes the interface to be brought down. For PPP encapsulation, five ignored keepalives causes the interface to be brought down. ECHOREQ packets are sent out only when LCP negotiation is complete (for example, when LCP is open).

Use the **keepalive** command in interface configuration mode to set the frequency at which LCP sends ECHOREQ packets to its peer. To restore the system to the default keepalive interval of 10 seconds, use the **keepalive** command with the **no** keyword. To disable keepalives, use the **keepalive disable** command. For both PPP and Cisco HDLC, a keepalive of 0 disables keepalives and is reported in the **show running-config** command output as **keepalive disable**.

When LCP is running on the peer and receives an ECHOREQ packet, it responds with an ECHOREP packet, regardless of whether keepalives are enabled on the peer.

Keepalives are independent between the two peers. One peer end can have keepalives enabled; the other end can have them disabled. Even if keepalives are disabled locally, LCP still responds with ECHOREP packets to the ECHOREQ packets it receives. Similarly, LCP also works if the period of keepalives at each end is different.

Frame Relay Encapsulation

When Frame Relay encapsulation is enabled on a serial interface, the interface configuration is hierarchical and comprises the following elements:

- The serial main interface comprises the physical interface and port. If you are not using the serial interface to support Cisco HDLC and PPP encapsulated connections, then you must configure subinterfaces with permanent virtual circuits (PVCs) under the serial main interface. Frame Relay connections are supported on PVCs only.
- Serial subinterfaces are configured under the serial main interface. A serial subinterface does not actively carry traffic until you configure a PVC under the serial subinterface. Layer 3 configuration typically takes place on the subinterface.
- When the encapsulation on a serial interface is changed from HDLC to any other encapsulation type, the configured serial subinterfaces on the main interface inherit the newly changed encapsulation and they do not get deleted.
- Point-to-point PVCs are configured under a serial subinterface. You cannot configure a PVC directly under a main interface. A single point-to-point PVC is allowed per subinterface. PVCs use a predefined circuit path and fail if the path is interrupted. PVCs remain active until the circuit is removed from either configuration. Connections on the serial PVC support Frame Relay encapsulation only.

Note The administrative state of a parent interface drives the state of the subinterface and its PVC. When the administrative state of a parent interface or subinterface changes, so does the administrative state of any child PVC configured under that parent interface or subinterface.

To configure Frame Relay encapsulation on serial interfaces, use the **encapsulation (Frame Relay VC-bundle)** command.

Frame Relay interfaces support two types of encapsulated frames:

- Cisco (default)
- IETF

Use the **encap** command in PVC configuration mode to configure Cisco or IETF encapsulation on a PVC. If the encapsulation type is not configured explicitly for a PVC, then that PVC inherits the encapsulation type from the main serial interface.

Note Cisco encapsulation is required on serial main interfaces that are configured for MPLS. IETF encapsulation is not supported for MPLS.

Before you configure Frame Relay encapsulation on an interface, you must verify that all prior Layer 3 configuration is removed from that interface. For example, you must ensure that there is no IP address configured directly under the main interface; otherwise, any Frame Relay configuration done under the main interface will not be viable.

LMI on Frame Relay Interfaces

The Local Management Interface (LMI) protocol monitors the addition, deletion, and status of PVCs. LMI also verifies the integrity of the link that forms a Frame Relay UNI interface. By default, **cisco** LMI is enabled on all PVCs.

If the LMI type is **cisco** (the default LMI type), the maximum number of PVCs that can be supported under a single interface is related to the MTU size of the main interface. Use the following formula to calculate the maximum number of PVCs supported on a card:

```
(MTU - 13)/8 = maximum number of PVCs
```

Note The default setting of the **mtu** command for a serial interface is 1504 bytes. Therefore, the default numbers of PVCs supported on a serial interface configured with **cisco** LMI is 186.

How to Configure Serial Interfaces

This section contains the following tasks:

- Configuring a Synchronous Serial Interface, page 45
- Configuring Asynchronous Serial Interface, page 46

Configuring a Synchronous Serial Interface

To configure a synchronous serial interface, perform the tasks in the following sections. Each task in the list is identified as either required or optional.

- Specifying a Synchronous Serial Interface, page 46 (Required)
- Specifying Synchronous Serial Encapsulation, page 46 (Optional)

Specifying a Synchronous Serial Interface

To specify a synchronous serial interface and enter interface configuration mode, use the following commands in global configuration mode.

Command	Purpose
Router(config)# **interface serial** *wic/slot/port* **Example:** `Router# interface serial 0/0/0`	Specifies the serial interface and enters interface configuration mode.

Specifying Synchronous Serial Encapsulation

By default, synchronous serial lines use the High-Level Data Link Control (HDLC) serial encapsulation method, which provides the synchronous framing and error detection functions of HDLC without windowing or retransmission. The serial interfaces support the following serial encapsulation methods:

- HDLC
- Frame Relay
- PPP

To define the encapsulation method, use the following command in interface configuration mode.

-f

Command	Purpose
Router(config-if)# **encapsulation** {**hdlc** | **frame-relay** | **ppp** } **Example:** `Router(config-if)# encapsulation ppp`	Configures synchronous serial encapsulation.

Encapsulation methods are set according to the type of protocol or application you configure in the Cisco IOS software.

For configuration examples, see the "Configuration Examples" section on page 4-47.

Configuring Asynchronous Serial Interface

You can use the **physical-layer async** command to change the interface mode from the default synchronous mode to asynchronous mode

SUMMARY STEPS

1. **physical-layer async**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**physical-layer async** **Example:** `Router(config-if)# physical-layer async`	Specifies the mode of a low-speed interface as either synchronous or asynchronous.

Note You cannot use the **physical-layer async** command for frame-relay encapsulation.

When you make a transition from asynchronous mode to synchronous mode in serial interfaces, the interface state becomes down by default. You should then use the **no shutdown** option to bring the interface up.

Configuration Examples

Example: PPP Configuration:

This example shows how to configure PPP encapsulation with CHAP authentication.

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
R1(config)# username R2 password cisco
R1(config)# interface serial 0/0/0
R1(config-if)# encapsulation ppp
R1(config-if)# ppp authentication chap
R1(config-if)# exit

Router> enable
Router# configure terminal
Router(config)# hostname R2
R2(config)# username R1 password  cisco
R2(config)# interface serial 0/0/0
R2(config-if)# encapsulation ppp
R2(config-if)# ppp authentication chap
R2(config-if)# exit
```

This example shows how to configure PPP encapsulation with PAP authentication.

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
R1(config)# username R2 password cisco
R1(config)# interface serial 0/0/0
R1(config-if)# encapsulation ppp
R1(config-if)# ppp authentication PAP
R1(config-if)# ppp pap sent-username R1 password cisco
R1(config-if)# end

Router> enable
Router#configure terminal
```

```
Router(config)# hostname R2
R2(config)# username R1 password cisco
R2(config)# interface serial 0/0/0
R2(config-if)# encapsulation ppp
R2(config-if)# ppp authentication PAP
R2(config-if)# ppp pap sent-username R2 password cisco
R2(config-if)# end
```

Example: Frame Relay Configuration

This example shows how to configure frame relay encapsulation on a serial interface.

```
Router1>enable
Router1#configure terminal
Router1(config)# interface Serial 0/0/0
Router1(config-if)# ip address 50.50.50.1 255.255.255.0
Router1(config-if)# encapsulation frame-relay
Router1(config-if)# no keepalive
Router1(config-if)# frame-relay interface-dlci 50
Router1(config-if)# end
Router2>enable
Router2#configure terminal
Router2(config)# interface Serial 0/2/0
Router2(config-if)# ip address 50.50.50.2 255.255.255.0
Router2(config-if)# encapsulation frame-relay
Router2(config-if)# no keepalive
Router2(config-if)# clock rate 2000000
Router2(config-if)# frame-relay interface-dlci 50
Router2(config-if)# end
```

Example: MLPPP Configuration

This example shows how to configure Multilink PPP on a serial interface.

```
Router1> enable
Router1# configure terminal
Router1(config)# interface Multilink 1
Router1(config-if)# ip address  120.120.120.1 255.255.255.0
Router1(config-if)# no ip route-cache
Router1(config-if)# ppp multilink
Router1(config-if)# ppp multilink group 1
Router1(config-if)# end
Router1(config)# interface Serial 0/2/0
Router1(config-if)# no ip address
Router1(config-if)# encapsulation ppp
Router1(config-if)# no ip route-cache
Router1(config-if)# ppp multilink
Router1(config-if)# ppp multilink group 1
Router1(config-if)# end
Router1(config)# interface Serial 0/2/1
Router1(config-if)# no ip address
Router1(config-if)# encapsulation ppp
Router1(config-if)# no ip route-cache
Router1(config-if)# ppp multilink
Router1(config-if)# ppp multilink group 1
Router1(config-if)# end
```

Example: Asynchronous Serial Configuration

This example shows how to configure a serial interface on asynchronous mode.

```
Router> enable
Router# configure terminal
Router(config)# interface serial 0/0/0
Router(config-if)# physical-layer async
Router(config-if)# ip address 10.0.0.2 255.0.0.0
Router(config-if)# async mode dedicated
Router(config-if)# end
```

Related Documents

Topic	Document Title
PPP and Multilink PPP	Configuring Media-Independent PPP and Multilink PPP
Serial Interface Configuration	Interface and Hardware Component Configuration Guide, Cisco IOS Release 15M&T
Frame Relay	Wide-Area Networking Configuration Guide: Frame Relay, Cisco IOS Release 15M&T

Configuring Ethernet Switch Ports

This chapter gives an overview of configuration tasks for the Gigabit Ethernet (GE) switch on the Cisco 800M Series ISR.

This chapter contains the following sections:

Configuring VLANs

A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment. Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in the VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router. A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment. Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in the VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router.

For detailed information on VLANs, see the following web link:

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750/software/release/15-0_2_se/configuration/guide/scg3750/swvlan.html

For a sample VLAN configuration, see "Example: VLAN configuration".

Example: VLAN configuration

The following example shows how to configure inter-VLAN routing:

```
Router# configure terminal
Router(config)# vlan 1
Router(config)# vlan 2
Router(config)# interface vlan 1
Router(config-if)# ip address 1.1.1.1 255.255.255.0
Router(config-if)# no shut
Router(config-if)# interface vlan 2
Roouter(config-if)# ip address 2.2.2.2 255.255.255.0
Router(config-if)# no shut
Router(config-if)# interface gigabitethernet 0/1
Router(config-if)# switchport access vlan 1
Router(config-if)# interface gigabitethernet 0/2
Router(config-if)# switchport access vlan 2
Router(config-if)# exit
```

Configuring VTP

VTP is a Layer 2 messaging protocol that maintains VLAN configuration consistency by managing the addition, deletion, and renaming of VLANs on a network-wide basis. VTP minimizes misconfigurations and configuration inconsistencies that can cause several problems, such as duplicate VLAN names, incorrect VLAN-type specifications, and security violations.

Before you create VLANs, you must decide whether to use VTP in your network. Using VTP, you can make configuration changes centrally on one or more switches and have those changes automatically communicated to all the other switches in the network. Without VTP, you cannot send information about VLANs to other switches.VTP is designed to work in an environment where updates are made on a single switch and are sent through VTP to other switches in the domain. It does not work well in a situation where multiple updates to the VLAN database occur simultaneously on switches in the same domain, which would result in an inconsistency in the VLAN database.

You should understand the following concepts for configuring VTP.

- VTP domain: A VTP domain (also called a VLAN management domain) consists of one switch or several interconnected switches or switch stacks under the same administrative responsibility sharing the same VTP domain name. A switch can be in only one VTP domain. You make global VLAN configuration changes for the domain.
- VTP server: In VTP server mode, you can create, modify, and delete VLANs, and specify other configuration parameters (such as the VTP version) for the entire VTP domain. VTP servers advertise their VLAN configurations to other switches in the same VTP domain and synchronize their VLAN configurations with other switches based on advertisements received over trunk links.VTP server is the default mode.
- VTP client: A VTP client behaves like a VTP server and transmits and receives VTP updates on its trunks, but you cannot create, change, or delete VLANs on a VTP client. VLANs are configured on another switch in the domain that is in server mode.
- VTP transparent: VTP transparent switches do not participate in VTP. A VTP transparent switch does not advertise its VLAN configuration and does not synchronize its VLAN configuration based on received advertisements. However, in VTP version 2 or version 3, transparent switches do forward VTP advertisements that they receive from other switches through their trunk interfaces. You can create, modify, and delete VLANs on a switch in VTP transparent mode.

For detailed information on VTP, see the following web link:

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750/software/release/15-0_2_se/configuration/guide/scg3750/swvtp.html

For a sample VTP configuration, see "Example: Configuring VTP".

Example: Configuring VTP

The following example shows how to configure the switch as a VTP server:

```
Router# configure terminal
Router(config)# vtp mode server
Router(config)# vtp domain Lab_Network
Router(config)# vtp password WATER
Router(config)# exit
```

The following example shows how to configure the switch as a VTP client:

```
Router# configure terminal
Router(config)# vtp mode client
Router(config)# exit
```

The following example shows how to configure the switch as VTP transparent:

```
Router# configure terminal
Router(config)# vtp mode transparent
Router# exit
```

Configuring 802.1x Authentication

IEEE 802.1x port-based authentication defines a client-server-based access control and authentication protocol to prevent unauthorized clients from connecting to a LAN through publicly accessible ports.The authentication server authenticates each client connected to a switch port before allowing access to any switch or LAN services. Until the client is authenticated, IEEE 802.1x access control allows only Extensible Authentication Protocol over LAN (EAPOL), Cisco Discovery Protocol (CDP), and Spanning Tree Protocol (STP) traffic through the port to which the client is connected. After authentication, normal traffic passes through the port.

With IEEE 802.1x authentication, the devices in the network have specific roles:

- Supplicant—Device (workstation) that requests access to the LAN and switch services and responds to requests from the router. The workstation must be running IEEE 802.1x-compliant client software such as that offered in the Microsoft Windows XP operating system. (The supplicant is sometimes called the client.)
- Authentication server—Device that performs the actual authentication of the supplicant. The authentication server validates the identity of the supplicant and notifies the router whether or not the supplicant is authorized to access the LAN and switch services. The Network Access Device (or Cisco ISR router in this instance) transparently passes the authentication messages between the supplicant and the authentication server, and the authentication process is carried out between the supplicant and the authentication server. The particular EAP method used will be decided between the supplicant and the authentication server (RADIUS server). The RADIUS security system with EAP extensions is available in Cisco Secure Access Control Server Version 3.0 or later. RADIUS operates in a client and server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.

- Authenticator—Router that controls the physical access to the network based on the authentication status of the supplicant. The router acts as an intermediary between the supplicant and the authentication server, requesting identity information from the supplicant, verifying that information with the authentication server, and relaying a response to the supplicant. The router includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

For detailed information on how to configure 802.1x port-based authentication, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_8021x/configuration/15-mt/sec-user-8021x-15-mt-book/config-ieee-802x-pba.html

For a sample 802.1x authentication configuration see "Example: Enabling IEEE 802.1x and AAA on a Switch Port".

Example: Enabling IEEE 802.1x and AAA on a Switch Port

This example shows how to configure Cisco 800M series ISR as 802.1x authenticator.

```
Router> enable
Router# configure terminal
Router(config)# dot1x system-auth-control
Router(config)# aaa new-model
Router(config)# aaa authentication dot1x default group radius
Router(config)# interface gigabitethernet 0/1
Router(config-if)# switchport mode access
Router(config-if)# authentication port-control auto
Router(config-if)# dot1x pae authenticator
Router(config-if)# end
```

Configuring Spanning Tree Protocol

Spanning Tree Protocol (STP) is a Layer 2 link management protocol that provides path redundancy while preventing loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. Multiple active paths among end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages. Switches might also learn end-station MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network. Spanning-tree operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

The STP uses a spanning-tree algorithm to select one switch of a redundantly connected network as the root of the spanning tree. The algorithm calculates the best loop-free path through a switched Layer 2 network by assigning a role to each port based on the role of the port in the active topology:

- Root—A forwarding port elected for the spanning-tree topology
- Designated—A forwarding port elected for every switched LAN segment
- Alternate—A blocked port providing an alternate path to the root bridge in the spanning tree
- Backup—A blocked port in a loopback configuration

The switch that has all of its ports as the designated role or as the backup role is the root switch. The switch that has at least one of its ports in the designated role is called the designated switch.Spanning tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the spanning-tree algorithm recalculates the spanning-tree topology and activates the standby path. Switches send and receive spanning-tree frames, called bridge protocol data units (BPDUs), at regular intervals. The switches do not forward these frames but use them to construct a loop-free path. BPDUs contain information about the sending switch and its ports, including

switch and MAC addresses, switch priority, port priority, and path cost. Spanning tree uses this information to elect the root switch and root port for the switched network and the root port and designated port for each switched segment.

When two ports on a switch are part of a loop, the spanning-tree port priority and path cost settings control which port is put in the forwarding state and which is put in the blocking state. The spanning-tree port priority value represents the location of a port in the network topology and how well it is located to pass traffic. The path cost value represents the media speed.

For detailed configuration information on STP see the following link:

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750/software/release/15-0_2_se/configuration/guide/scg3750/swstp.html

For configuration examples, see "Example: Spanning Tree Protocol Configuration".

Example: Spanning Tree Protocol Configuration

The following example shows configuring spanning-tree port priority of a Gigabit Ethernet interface. If a loop occurs, spanning tree uses the port priority when selecting an interface to put in the forwarding state.

```
Router# configure terminal
Router(config)# interface gigabitethernet 0/2
Router(config-if)# spanning-tree vlan 1 port-priority 64
Router(config-if)# end
```

The following example shows how to change the spanning-tree port cost of a Gigabit Ethernet interface. If a loop occurs, spanning tree uses cost when selecting an interface to put in the forwarding state.

```
Router#configure terminal
Router(config)# interface gigabitethernet 0/2
Router(config-if)# spanning-tree cost 18
Router(config-if)# end
```

The following example shows configuring the bridge priority of VLAN 10 to 33792:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 priority 33792
Router(config)# end
```

The following example shows configuring the hello time for VLAN 10 being configured to 7 seconds. The hello time is the interval between the generation of configuration messages by the root switch.

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 hello-time 4
Router(config)# end
```

The following example shows configuring forward delay time. The forward delay is the number of seconds an interface waits before changing from its spanning-tree learning and listening states to the forwarding state.

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 forward-time 21
Router(config)# end
```

The following example shows configuring maximum age interval for the spanning tree. The maximum-aging time is the number of seconds a switch waits without receiving spanning-tree configuration messages before attempting a reconfiguration.

```
Router# configure terminal
Router(config)# spanning-tree vlan 20 max-age 36
Router(config)# end
```

The following example shows the switch being configured as the root bridge for VLAN 10, with a network diameter of 4.

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 root primary diameter 4
Router(config)# exit
```

Configuring MAC Address Table Manipulation

The MAC address table contains address information that the switch uses to forward traffic between ports. All MAC addresses in the address table are associated with one or more ports. The address table includes these types of addresses:

- Dynamic address: a source MAC address that the switch learns and then drops when it is not in use. You can use the aging time setting to define how long the switch retains unseen addresses in the table.
- Static address: a manually entered unicast address that does not age and that is not lost when the switch resets.

The address table lists the destination MAC address, the associated VLAN ID, and port number associated with the address and the type (static or dynamic).

See the "Example: MAC Address Table Manipulation" for sample configurations for enabling secure MAC address, creating a statc entry, set the maximum number of secure MAC addresses and set the aging time.

For detailed configuration information on MAC address table manipulation see the following link:

http://www.cisco.com/c/en/us/td/docs/routers/access/interfaces/software/feature/guide/geshwic_cfg.html#wp1048223

Example: MAC Address Table Manipulation

The following example shows configuration for enabling secure MAC address option on the port.

```
Router# configure terminal
Router(config)# mac-address-table secure 0004.0005.0006 GigabitEthernet 0/1 vlan 5
Router(config)# end
```

The following example shows creating a static entry in the MAC address table.

```
Router# configure terminal
Router(config)# mac-address-table static 0002.0003.0004 interface GigabitEthernet 0/2 vlan
3
Router(config)# end
```

The following example sets the maximum number of secure MAC addresses to 10.

```
Router# configure terminal
Router(config)# mac-address-table secure maximum 10 GigabitEthernet 0/1
Router(config)# end
```

The following example shows setting the aging timer.

```
Router# configure terminal
Router(config)# mac-address-table aging-time 300
```

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

Configuring MAC Address Notification Traps

MAC address notification enables you to track users on a network by storing the MAC address activity on the switch. Whenever the switch learns or removes a MAC address, an SNMP notification can be generated and sent to the network management system (NMS). If you have many users coming and going from the network, you can set a trap interval time to bundle the notification traps and reduce network traffic. The MAC notification history table stores the MAC address activity for each hardware port for which the trap is enabled. MAC address notifications are generated for dynamic and secure MAC addresses; events are not generated for self addresses, multicast addresses, or other static addresses.

For configuration examples, see "Example: Configuring MAC Address Notification Traps".

Example: Configuring MAC Address Notification Traps

This example shows how to enable the MAC notification trap when a MAC address is added to the interface:

```
Router(config)# interface gigabitethernet 0/1
Router(config-if)# snmp trap mac-notification added
Router(config-if)# end
```

This example shows how to enable the MAC notification trap when a MAC address is removed from this interface.

```
Router(config)# interface gigabitethernet 0/1
Router(config-if)# snmp trap mac-notification removed
Router(config-if)# end
```

Configuring the Switched Port Analyzer

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

Only traffic that enters or leaves source ports or traffic that enters or leaves source VLANs can be monitored by using SPAN; traffic routed to a source VLAN cannot be monitored. For example, if incoming traffic is being monitored, traffic that gets routed from another VLAN to the source VLAN cannot be monitored; however, traffic that is received on the source VLAN and routed to another VLAN can be monitored.

See Example: SPAN Configuration, page 58 for SPAN configuration examples.

For detailed information on how to configure a switched port analyzer (SPAN) session, see the following web link:

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750/software/release/15-0_2_se/configuration/guide/scg3750/swspan.html

Example: SPAN Configuration

The following example shows how to configure a SPAN session to monitor bidirectional traffic from a Gigabit Ethernet source interface:

```
Router# configure terminal
Router(config)# monitor session 1 source gigabitethernet 0/1
Router(config)# end
```

The following example shows how to configure a gigabit ethernet interface as the destination for a SPAN session:

```
Router# configure terminal
Router(config)# monitor session 1 destination gigabitethernet 0/2
Router(config)# end
```

The following example shows how to remove gigabit ethernet as a SPAN source for SPAN session 1:

```
Router# configure terminal
Router(config)# no monitor session 1 source gigabitethernet 0/1
Router(config)# end
```

Configuring IGMP Snooping

IGMP snooping constrains the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. As the name implies, IGMP snooping requires the LAN switch to snoop on the IGMP transmissions between the host and the router and to keep track of multicast groups and member ports. When the switch receives an IGMP report from a host for a particular multicast group, the switch adds the host port number to the forwarding table entry; when it receives an IGMP Leave Group message from a host, it removes the host port from the table entry. It also periodically deletes entries if it does not receive IGMP membership reports from the multicast clients.

The multicast router sends out periodic general queries to all VLANs. All hosts interested in this multicast traffic send join requests and are added to the forwarding table entry. The switch creates one entry per VLAN in the IGMP snooping IP multicast forwarding table for each group from which it receives an IGMP join request.

By default, IGMP snooping is globally enabled. When globally enabled or disabled, it is also enabled or disabled in all existing VLAN interfaces. By default, IGMP snooping is enabled on all VLANs, but it can be enabled and disabled on a per-VLAN basis. Global IGMP snooping overrides the per-VLAN IGMP snooping capability. If global snooping is disabled, you cannot enable VLAN snooping. If global snooping is enabled, you can enable or disable snooping on a VLAN basis.

See the "Example: Configuring IGMP Snooping" for a sample configuration on IGMP snooping.

Example: Configuring IGMP Snooping

The following example shows how to enable IGMP snooping on a VLAN interface.

```
Router# configure terminal
Router(config)# ip igmp snooping vlan 1
Router# end
```

The following example shows how to enable a static connection to a multicast router.

```
Router# configure terminal
Router(config)# ip igmp snooping vlan 1 mrouter interface gigabitethernet 0/1
Router# end
```

The following example shows how to add a port as a member of a multicast group. Ports normally join multicast groups through the IGMP report message, but you can also statically configure a port as a member of a multicast group.

```
Router# configure terminal
Router(config)# ip igmp snooping vlan 1 static 0100.5e02.0203 interface gigabitethernet
0/1
Router# end
```

Configuring Per-Port Storm Control

Storm control prevents traffic on a LAN from being disrupted by a broadcast, a multicast, or a unicast storm on one of the physical interfaces. A LAN storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. Errors in the protocol-stack implementation, mistakes in the network configuration, or users issuing a denial-of-service attack can cause a storm.

Storm control (or traffic suppression) monitors packets passing from an interface to the switching bus and determines if the packet is unicast, multicast, or broadcast. The switch counts the number of packets of a specified type received within the 1-second time interval and compares the measurement with a predefined suppression-level threshold.

Storm control uses one of these methods to measure traffic activity:

- Bandwidth as a percentage of the total available bandwidth of the port that can be used by the broadcast, multicast, or unicast traffic
- Traffic rate in packets per second at which broadcast, multicast, or unicast packets are received

With either method, the port blocks traffic when the rising threshold is reached. The port remains blocked until the traffic rate drops below the falling threshold (if one is specified) and then resumes normal forwarding. If the falling suppression level is not specified, the switch blocks all traffic until the traffic rate drops below the rising suppression level. In general, the higher the level, the less effective the protection against broadcast storms.

Note In C800M platform, when you configure the **storm-control action shutdown** command, the state of the port changes to administratively down. Use the **no shutdown** command to manually revert the state of the port.

See the "Example: Per-Port Storm-Control" for a sample configuration on per-port storm control.

Example: Per-Port Storm-Control

The following example shows bandwidth-based multicast storm control being enabled at 70 percent on Gigabit Ethernet interface.

```
Router# configure terminal
Router(config)# interface gigabitethernet 0/2
Router(config-if)# storm-control multicast level 70.0 30.0
Router(config-if)# end
Router# show storm-control multicast
Interface  Filter State   Upper    Lower    Current
---------  -------------  -------  -------  -------
Gi0/0      inactive       100.00%  100.00%      N/A
Gi0/1      inactive       100.00%  100.00%      N/A
Gi0/2      Forwarding      70.00%   30.00%    0.00%
```

Configuring HSRP

The Hot Standby Router Protocol (HSRP) is Cisco's standard method of providing high network availability by providing first-hop redundancy for IP hosts on an IEEE 802 LAN configured with a default gateway IP address. HSRP routes IP traffic without relying on the availability of any single router. It enables a set of router interfaces to work together to present the appearance of a single virtual router or default gateway to the hosts on a LAN. When HSRP is configured on a network or segment, it provides a virtual Media Access Control (MAC) address and an IP address that is shared among a group of configured routers. HSRP allows two or more HSRP-configured routers to use the MAC address and IP network address of a virtual router. The virtual router does not exist; it represents the common target for routers that are configured to provide backup to each other. One of the routers is selected to be the active router and another to be the standby router, which assumes control of the group MAC address and IP address should the designated active router fail.

HSRP uses a priority mechanism to determine which HSRP configured device is to be the default active device. To configure a device as the active device, you assign it a priority that is higher than the priority of all the other HSRP-configured devices. The default priority is 100, so if you configure just one device to have a higher priority, that device will be the default active device. In case of ties, the primary IP addresses are compared, and the higher IP address has priority. If you do not use the standby preempt interface configuration command in the configuration for a router, that router will not become the active router, even if its priority is higher than all other routers.

For more information about configuring HSRP, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipapp_fhrp/configuration/15-mt/fhp-15-mt-book/fhp-hsrp.html

For a sample HSRP configuration, see "Example: Configuring HSRP"

Example: Configuring HSRP

In this example, Router A is configured to be the active device for group 1 and standby device for group 2. Device B is configured as the active device for group 2 and standby device for group 1.

```
RouterA# configure terminal
RouterA(config)# interface GigabitEthernet 0/1
RouterA(config-if)# ip address 10.1.0.21 255.255.0.0
RouterA(config-if)# standby 1 priority 110
RouterA(config-if)# standby 1 preempt
RouterA(config-if)# standby 1 ip 10.1.0.3
RouterA(config-if)# standby 2 priority 95
RouterA(config-if)# standby 2 preempt
RouterA(config-if)# standby 2 ip 10.1.0.4
RouterA(config-if)# end

RouterB# configure terminal
RouterB(config)# interface GigabitEthernet 0/1
RouterB(config-if)# ip address 10.1.0.22 255.255.0.0
RouterB(config-if)# standby 1 priority 105
RouterB(config-if)# standby 1 preempt
RouterB(config-if)# standby 1 ip 10.1.0.3
RouterB(config-if)# standby 2 priority 110
RouterB(config-if)# standby 2 preempt
RouterB(config-if)# standby 2 ip 10.1.0.4
```

Configuring VRRP

The Virtual Router Redundancy Protocol (VRRP) is an election protocol that dynamically assigns responsibility for one or more virtual routers to the VRRP routers on a LAN, allowing several routers on a multiaccess link to utilize the same virtual IP address. A VRRP router is configured to run the VRRP protocol in conjunction with one or more other routers attached to a LAN. In a VRRP configuration, one router is elected as the virtual router master, with the other routers acting as backups in case the virtual router master fails.

An important aspect of the VRRP is VRRP router priority. Priority determines the role that each VRRP router plays and what happens if the virtual router master fails. If a VRRP router owns the IP address of the virtual router and the IP address of the physical interface, this router will function as a virtual router master. Priority also determines if a VRRP router functions as a virtual router backup and the order of ascendancy to becoming a virtual router master if the virtual router master fails. You can configure the priority of each virtual router backup using the **vrrp priority** command.

By default, a preemptive scheme is enabled whereby a higher priority virtual router backup that becomes available takes over for the virtual router backup that was elected to become virtual router master. You can disable this preemptive scheme using the **no vrrp preempt** command. If preemption is disabled, the virtual router backup that is elected to become virtual router master remains the master until the original virtual router master recovers and becomes master again.

The virtual router master sends VRRP advertisements to other VRRP routers in the same group. The advertisements communicate the priority and state of the virtual router master. The VRRP advertisements are encapsulated in IP packets and sent to the IP Version 4 multicast address assigned to the VRRP group. The advertisements are sent every second by default; the interval is configurable.

For more information on VRRP, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipapp_fhrp/configuration/15-mt/fhp-15-mt-book/fhp-vrrp.html

For a sample VRRP configuration, see “Example: Configuring VRRP”.

Example: Configuring VRRP

In the following example, Router A and Router B each belong to two VRRP groups, group1 and group 5. In this configuration, each group has the following properties:

Group 1:

- Virtual IP address is 10.1.0.10.
- Router A will become the master for this group with priority 120.
- Advertising interval is 3 seconds.
- Preemption is enabled.

Group 5:

- Router B will become the master for this group with priority 200.
- Advertising interval is 30 seconds.
- Preemption is enabled.

```
RouterA(config)# interface GigabitEthernet 0/1
RouterA(config-if)# ip address 10.1.0.2 255.0.0.0
RouterA(config-if)# vrrp 1 priority 120
RouterA(config-if)# vrrp 1 authentication cisco
RouterA(config-if)# vrrp 1 timers advertise 3
```

```
RouterA(config-if)# vrrp 1 timers learn
RouterA(config-if)# vrrp 1 ip 10.1.0.10
RouterA(config-if)# vrrp 5 priority 100
RouterA(config-if)# vrrp 5 timers advertise 30
RouterA(config-if)# vrrp 5 timers learn
RouterA(config-if)# vrrp 5 ip 10.1.0.50
RouterA(config-if)# no shutdown

RouterA(config-if)# end

RouterB(config)# interface GigabitEthernet 0/1
RouterB(config-if)# ip address 10.1.0.1 255.0.0.0
RouterB(config-if)# vrrp 1 priority 100
RouterB(config-if)# vrrp 1 authentication cisco
RouterB(config-if)# vrrp 1 timers advertise 3
RouterB(config-if)# vrrp 1 timers learn
RouterB(config-if)# vrrp 1 ip 10.1.0.10
RouterB(config-if)# vrrp 5 priority 200
RouterB(config-if)# vrrp 5 timers advertise 30
RouterB(config-if)# vrrp 5 timers learn
RouterB(config-if)# vrrp 5 ip 10.1.0.50
RouterB(config-if)# no shutdown
RouterB(config-if)# end
```

Configuring Security Features

The Cisco 800M Series ISR provides the following security features:

Configuring Authentication, Authorization, and Accounting

Authentication, Authorization, and Accounting (AAA) network security services provide the primary framework through which you set up access control on your router. Authentication provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and encryption depending on the security protocol you choose. Authorization provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, Internetwork Packet Exchange (IPX), AppleTalk Remote Access (ARA), and Telnet. Accounting provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

AAA uses protocols such as Remote Authentication Dial-In User Service (RADIUS), Terminal Access Controller Access Control System Plus (TACACS+), or Kerberos to administer its security functions. If your router is acting as a network access server, AAA is the means through which you establish communication between your network access server and your RADIUS, TACACS+, or Kerberos security server.

For information about configuring AAA services and supported security protocols, see the following guide:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_aaa/configuration/15-mt/sec-usr-aaa-15-mt-book.html

Configuring Access Lists

Access lists permit or deny network traffic over an interface, based on source IP address, destination IP address, or protocol. Access lists are configured as standard or extended. A standard access list either permits or denies passage of packets from a designated source. An extended access list allows designation of both the destination and the source, and it allows designation of individual protocols to be permitted or denied passage.

An access list is a series of commands with a common tag to bind them together. The tag is either a number or a name. Table 6-1 lists the commands used to configure access lists.

Table 6-1 Access List Configuration Commands

Access Control List (ACL) Type	Configuration Commands
Numbered	
Standard	**access-list** {**1-99**} {**permit** | **deny**} *source-addr* [*source-mask*]
Extended	**access-list** {**100-199**} {**permit** | **deny**} *protocol source-addr* [*source-mask*] *destination-addr* [*destination-mask*]
Named	
Standard	**ip access-list standard** *name* **deny** {*source* | *source-wildcard* | **any**}
Extended	**ip access-list extended** *name* {**permit** | **deny**} *protocol* {*source-addr* [*source-mask*] | **any**} {*destination-addr* [*destination-mask*] | **any**}

For more complete information on creating access lists, see the following web link:
http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_acl/configuration/15-mt/sec-data-acl-15-mt-book.html

Access Groups

An access group is a sequence of access list definitions bound together with a common name or number. An access group is enabled for an interface during interface configuration. Use the following guidelines when creating access groups:

- The order of access list definitions is significant. A packet is compared against the first access list in the sequence. If there is no match (that is, if neither a permit nor a deny occurs), the packet is compared with the next access list, and so on.
- All parameters must match the access list before the packet is permitted or denied.
- There is an implicit “deny all” at the end of all sequences.

For information on configuring and managing access groups, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_acl/configuration/15-mt/sec-data-acl-15-mt-book/sec-create-ip-al-filter.html

Configuring Cisco IOS IPS

The Cisco IOS Intrusion Prevention System (IPS) acts as an in-line intrusion detection sensor, watching packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When Cisco IOS IPS detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or Security Device Event Exchange (SDEE). The network administrator can configure Cisco IOS IPS to choose the appropriate response to various threats. When packets in a session match a signature, Cisco IOS IPS can take any of the following actions, as appropriate:

- Send an alarm to a syslog server or a centralized management interface
- Drop the packet
- Reset the connection
- Deny traffic from the source IP address of the attacker for a specified amount of time
- Deny traffic on the connection for which the signature was seen for a specified amount of time

For more information about configuring Cisco IOS IPS see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios/sec_data_plane/configuration/guide/convert/sec_data_ios_ips_15_1_book/sec_cfg_ips.html

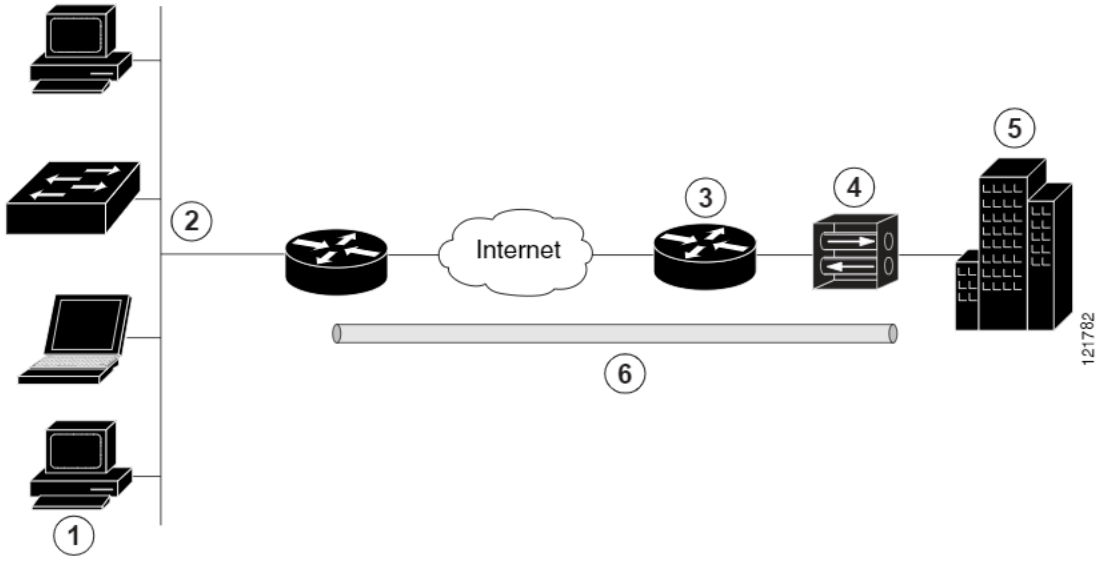
Configuring VPN

A Virtual Private Network (VPN) connection provides a secure connection between two networks over a public network such as the Internet. Cisco 800M Series ISRs support two types of VPNs: site-to-site and remote access. Remote access VPNs are used by remote clients to log in to a corporate network. Site-to-site VPNs connect branch offices to corporate offices. This section gives examples for site-to-site and remote access VPNs.

Remote Access VPN Example

The configuration of a remote access VPN uses Cisco Easy VPN and an IP Security (IPSec) tunnel to configure and secure the connection between the remote client and the corporate network. Figure 6-1 shows a typical deployment scenario.

Figure 6-1 Remote Access VPN Using IPSec Tunnel



1	Remote networked users
2	VPN client—Cisco 800M Series ISR
3	Router—Provides corporate office network access
4	VPN server—Easy VPN server; for example, a Cisco VPN 3000 concentrator with outside interface address 210.110.101.1
5	Corporate office with a network address of 10.1.1.1
6	IPSec tunnel

The Cisco Easy VPN client feature eliminates much of the tedious configuration work by implementing the Cisco Unity Client protocol. This protocol allows most VPN parameters, such as internal IP addresses, internal subnet masks, DHCP server addresses, Windows Internet Naming Service (WINS) server addresses, and split-tunneling flags, to be defined at a VPN server, such as a Cisco VPN 3000 series concentrator that is acting as an IPSec server.

A Cisco Easy VPN server–enabled device can terminate VPN tunnels initiated by mobile and remote workers who are running Cisco Easy VPN Remote software on PCs. Cisco Easy VPN server–enabled devices allow remote routers to act as Cisco Easy VPN Remote nodes.

The Cisco Easy VPN client feature can be configured in one of two modes—client mode or network extension mode. Client mode is the default configuration and allows only devices at the client site to access resources at the central site. Resources at the client site are unavailable to the central site. Network extension mode allows users at the central site (where the Cisco VPN 3000 series concentrator is located) to access network resources on the client site.

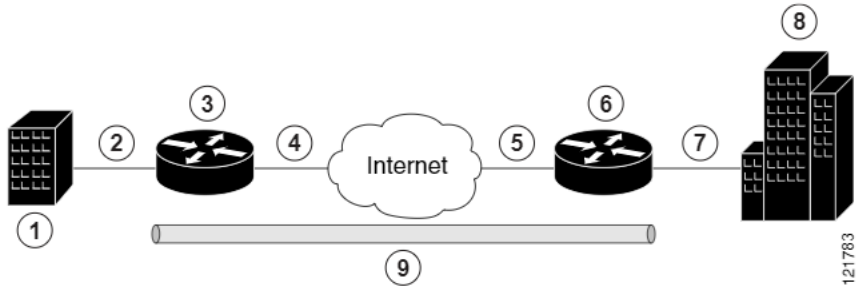
After the IPSec server has been configured, a VPN connection can be created with minimal configuration on an IPSec client. When the IPSec client initiates the VPN tunnel connection, the IPSec server pushes the IPSec policies to the IPSec client and creates the corresponding VPN tunnel connection.

Note The Cisco Easy VPN client feature supports configuration of only one destination peer. If your application requires creation of multiple VPN tunnels, you must manually configure the IPSec VPN and Network Address Translation/Peer Address Translation (NAT/PAT) parameters on both the client and the server.

Site-to-Site VPN Example

The configuration of a site-to-site VPN uses IPSec and the generic routing encapsulation (GRE) protocol to secure the connection between the branch office and the corporate network. Figure 6-2 shows a typical deployment scenario.

Figure 6-2 Site-to-Site VPN Using an IPSec Tunnel and GRE



1	Branch office containing multiple LANs and VLANs
2	Gigabit Ethernet LAN interface—With address 192.165.0.0/16 (also the inside interface for NAT)
3	VPN client—Cisco 800M Series ISR
4	Gigabit Ethernet interface—With address 200.1.1.1 (also the outside interface for NAT)
5	LAN interface—Connects to the Internet; with outside interface address of 210.110.101.1
6	VPN client—Another router, which controls access to the corporate network
7	LAN interface—Connects to the corporate network; with inside interface address of 10.1.1.1
8	Corporate office network
9	IPSec tunnel with GRE

For more information about IPSec and GRE configuration, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_vpnips/configuration/15-mt/sec-sec-for-vpns-w-ipsec-15-mt-book/sec-cfg-vpn-ipsec.html

Configuration Examples

Each example configures a VPN over an IPSec tunnel, using the procedure given in the "Configure a VPN over an IPSec Tunnel" section on page 68. Then, the specific procedure for a remote access configuration is given, followed by the specific procedure for a site-to-site configuration.

The examples shown in this chapter apply only to the endpoint configuration on the Cisco 800M Series ISRs. Any VPN connection requires both endpoints to be properly configured in order to function. See the software configuration documentation as needed to configure VPN for other router models.

VPN configuration information must be configured on both endpoints. You must specify parameters such as internal IP addresses, internal subnet masks, DHCP server addresses, and Network Address Translation (NAT).

- "Configure a VPN over an IPSec Tunnel" section on page 68
- "Create a Cisco Easy VPN Remote Configuration" section on page 77
- "Configure a Site-to-Site GRE Tunnel" section on page 80

Configure a VPN over an IPSec Tunnel

Perform the following tasks to configure a VPN over an IPSec tunnel:

- Configure the IKE Policy, page 69
- Configure Group Policy Information, page 70
- Apply Mode Configuration to the Crypto Map, page 72
- Enable Policy Lookup, page 73
- Configure IPSec Transforms and Protocols, page 74
- Configure the IPSec Crypto Method and Parameters, page 75
- Apply the Crypto Map to the Physical Interface, page 76
- Where to Go Next, page 77

Configure the IKE Policy

To configure the Internet Key Exchange (IKE) policy, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **crypto isakmp policy** *priority*
2. **encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**}
3. **hash** {**md5** | **sha**}
4. **authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
5. **group** {**1** | **2** | **5**}
6. **lifetime** *seconds*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 1`	Creates an IKE policy that is used during IKE negotiation. The priority is a number from 1 to 10000, with 1 being the highest. Also enters the ISAKMP[1] policy configuration mode.
Step 2	**encryption** {**des**	**3des**	**aes**	**aes 192**	**aes 256**} **Example:** `Router(config-isakmp)# encryption 3des`	Specifies the encryption algorithm used in the IKE policy. The example specifies 168-bit DES[2].
Step 3	**hash** {**md5**	**sha**} **Example:** `Router(config-isakmp)# hash md5`	Specifies the hash algorithm used in the IKE policy. The example specifies the MD5[3] algorithm. The default is SHA-1[4].
Step 4	**authentication** {**rsa-sig**	**rsa-encr**	**pre-share**} **Example:** `Router(config-isakmp)# authentication pre-share`	Specifies the authentication method used in the IKE policy. The example specifies a pre-shared key.
Step 5	**group** {**1**	**2**	**5**} **Example:** `Router(config-isakmp)# group 2`	Specifies the Diffie-Hellman group to be used in an IKE policy.

	Command or Action	Purpose
Step 6	**lifetime** *seconds* **Example:** `Router(config-isakmp)# lifetime 480`	Specifies the lifetime, from 60 to 86400 seconds, for an IKE SA[5].
Step 7	**exit** **Example:** `Router(config-isakmp)# exit`	Exits IKE policy configuration mode and enters global configuration mode.

1. ISAKMP = Internet Security Association Key and Management Protocol
2. DES = data encryption standard
3. MD5 = Message Digest 5
4. SHA-1 = Secure Hash standard
5. SA = security association

Configure Group Policy Information

To configure the group policy, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **crypto isakmp client configuration group** {*group-name* | *default*}
2. **key** ***name***
3. **dns** ***primary-server***
4. **domain** ***name***
5. **exit**
6. **ip local pool {default | *poolname*} [*low-ip-address* [*high-ip-address*]]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto isakmp client configuration group** {*group-name* | *default*} **Example:** `Router(config)# crypto isakmp client configuration group rtr-remote`	Creates an IKE policy group containing attributes to be downloaded to the remote client. Also enters the ISAKMP group policy configuration mode.
Step 2	**key** *name* **Example:** `Router(config-isakmp-group)# key secret-password`	Specifies the IKE pre-shared key for the group policy.

	Command or Action	Purpose
Step 3	**dns** *primary-server* **Example:** `Router(config-isakmp-group)# dns 10.50.10.1`	Specifies the primary DNS[1] server for the group. You may also want to specify WINS[2] servers for the group by using the **wins** command.
Step 4	**domain** *name* **Example:** `Router(config-isakmp-group)# domain company.com`	Specifies group domain membership.
Step 5	**exit** **Example:** `Router(config-isakmp-group)# exit`	Exits IKE group policy configuration mode and enters global configuration mode.
Step 6	**ip local pool** {**default** | *poolname*} [*low-ip-address* [*high-ip-address*]] **Example:** `Router(config)# ip local pool dynpool 30.30.30.20 30.30.30.30`	Specifies a local address pool for the group. For details about this command and additional parameters that can be set, see *Cisco IOS Dial Technologies Command Reference*.

1. DNS = Domain Name System
2. WINS = Windows Internet Naming Service

Apply Mode Configuration to the Crypto Map

To apply mode configuration to the crypto map, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **crypto map** *map-name* **isakmp authorization list** *list-name*
2. **crypto map** *tag* **client configuration address** [**initiate** | **respond**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto map** *map-name* **isakmp authorization list** *list-name* **Example:** `Router(config)# crypto map dynmap isakmp authorization list rtr-remote`	Applies mode configuration to the crypto map and enables key lookup (IKE queries) for the group policy from an AAA server.

Step 2	**crypto map** *tag* **client configuration address** [**initiate**	**respond**]
		
		Example:
		`Router(config)# crypto map dynmap client configuration address respond`
		`#`
			Configures the router to reply to mode configuration requests from remote clients.

Enable Policy Lookup

To enable policy lookup through AAA, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **aaa new-model**
2. **aaa authentication login** {**default** | *list-name*} *method1* [*method2...*]
3. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [*method1* [*method2...*]]
4. **username** *name* {**nopassword** | **password** *password* | **password** *encryption-type encrypted-password*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 2	**aaa authentication login** {**default**	*list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authentication login rtr-remote local`	Specifies AAA authentication of selected users at login, and specifies the method used. This example uses a local authentication database. You could also use a RADIUS server for this. For details, see *Cisco IOS Security Configuration Guide: Securing User Services, Release 15M&T* and *Cisco IOS Security Command Reference*.
Step 3	**aaa authorization** {**network**	**exec**	**commands** *level*	**reverse-access**	**configuration**} {**default**	*list-name*} [*method1* [*method2...*]] **Example:** `Router(config)# aaa authorization network rtr-remote local`	Specifies AAA authorization of all network-related service requests, including PPP, and specifies the method of authorization.
Step 4	**username** *name* {**nopassword**	**password** *password*	**password** *encryption-type encrypted-password*} **Example:** `Router(config)# username username1 password 0 password1`	Establishes a username-based authentication system.

Configure IPSec Transforms and Protocols

A transform set represents a certain combination of security protocols and algorithms. During IKE negotiation, the peers agree to use a particular transform set for protecting data flow.

During IKE negotiations, the peers search multiple transform sets for a transform that is the same at both peers. When a transform set is found that contains such a transform, it is selected and applied to the protected traffic as a part of both peers' configurations.

To specify the IPSec transform set and protocols, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **crypto ipsec profile** *profile-name*
2. **crypto ipsec transform-set** *transform-set-name*
3. **crypto ipsec security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*}

DETAILED STEPS

	Command or Action	Purpose

Step 1	**crypto ipsec profile** *profile-name*
	
	Example:
	`Router(config)# crypto ipsec profile pro1`
	`Router(config)#`
		Configures an IPSec profile to apply protection on the tunnel for encryption.

Step 2	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*] **Example:** `Router(config)# crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac`	Defines a transform set—an acceptable combination of IPSec security protocols and algorithms. See *Cisco IOS Security Command Reference* for detail about the valid transforms and combinations.
Step 3	**crypto ipsec security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*} **Example:** `Router(config)# crypto ipsec security-association lifetime seconds 86400`	Specifies global lifetime values used when IPSec security associations are negotiated.

Configure the IPSec Crypto Method and Parameters

A dynamic crypto map policy processes negotiation requests for new security associations from remote IPSec peers, even if the router does not know all the crypto map parameters (for example, IP address).

To configure the IPSec crypto method, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **crypto dynamic-map** *dynamic-map-name dynamic-seq-num*
2. **set transform-set** ***transform-set-name*** **[*transform-set-name2...transform-set-name6*]**
3. **reverse-route**
4. **exit**
5. **crypto map** ***map-name seq-num*** **[ipsec-isakmp] [dynamic** ***dynamic-map-name***] **[discover] [profile** ***profile-name***]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **Example:** `Router(config)# crypto dynamic-map dynmap 1`	Creates a dynamic crypto map entry and enters crypto map configuration mode. See *Cisco IOS Security Command Reference* for more detail about this command.
Step 2	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config-crypto-map)# set transform-set vpn1`	Specifies which transform sets can be used with the crypto map entry.
Step 3	**reverse-route** **Example:** `Router(config-crypto-map)# reverse-route`	Creates source proxy information for the crypto map entry.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Router(config-crypto-map)# exit`	Returns to global configuration mode.
Step 5	**crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*] **Example:** `Router(config)# crypto map static-map 1 ipsec-isakmp dynamic dynmap`	Creates a crypto map profile.

Apply the Crypto Map to the Physical Interface

The crypto maps must be applied to each interface through which IPSec traffic flows. Applying the crypto map to the physical interface instructs the router to evaluate all the traffic against the security associations database. With the default configurations, the router provides secure connectivity by encrypting the traffic sent between remote sites. However, the public interface still allows the rest of the traffic to pass and provides connectivity to the Internet.

To apply a crypto map to an interface, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **interface** *type number*
2. **crypto map** *map-name*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0`	Enters the interface configuration mode for the interface to which you are applying the crypto map.

	Command or Action	Purpose
Step 2	**crypto map** *map-name* **Example:** `Router(config-if)# crypto map static-map`	Applies the crypto map to the interface.
Step 3	**exit** **Example:** `Router(config-crypto-map)# exit`	Returns to global configuration mode.

Where to Go Next

If you are creating a Cisco Easy VPN remote configuration, go to the "Create a Cisco Easy VPN Remote Configuration" section on page 77.

If you are creating a site-to-site VPN using IPSec tunnels and GRE, go to the "Configure a Site-to-Site GRE Tunnel" section on page 80.

Create a Cisco Easy VPN Remote Configuration

The router that is acting as the Cisco Easy VPN client must create a Cisco Easy VPN remote configuration and assign it to the outgoing interface.

To create the remote configuration, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **crypto ipsec client ezvpn** *name*
2. **group** *group-name* **key** *group-key*
3. **peer** {*ipaddress* | *hostname*}
4. **mode** {**client** | **network-extension** | **network extension plus**}
5. **exit**
6. **crypto isakmp keepalive** *seconds*
7. **interface** *type number*
8. **crypto ipsec client ezvpn** *name* [**outside** | **inside**]
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto ipsec client ezvpn** *name* **Example:** `Router(config)# crypto ipsec client ezvpn ezvpnclient`	Creates a Cisco Easy VPN remote configuration, and enters Cisco Easy VPN remote configuration mode.
Step 2	**group** *group-name* **key** *group-key* **Example:** `Router(config-crypto-ezvpn)# group ezvpnclient key secret-password`	Specifies the IPSec group and IPSec key value for the VPN connection.
Step 3	**peer** {*ipaddress* | *hostname*} **Example:** `Router(config-crypto-ezvpn)# peer 192.168.100.1`	Specifies the peer IP address or hostname for the VPN connection. **Note** A hostname can be specified only when the router has a DNS server available for hostname resolution. **Note** Use this command to configure multiple peers for use as backup. If one peer goes down, the Easy VPN tunnel is established with the second available peer. When the primary peer comes up again, the tunnel is reestablished with the primary peer.
Step 4	**mode** {**client** | **network-extension** | **network extension plus**} **Example:** `Router(config-crypto-ezvpn)# mode client`	Specifies the VPN mode of operation.
Step 5	**exit** **Example:** `Router(config-crypto-ezvpn)# exit`	Returns to global configuration mode.
Step 6	**crypto isakmp keepalive** *seconds* **Example:** `Router(config-crypto-ezvpn)# crypto isakmp keepalive 10`	Enables dead peer detection messages. Time between messages is given in seconds, with a range of 10 to 3600.

	Command or Action	Purpose
Step 7	**interface** *type number* **Example:** `Router(config)# interface Gigabitethernet 0/2`	Enters the interface configuration mode for the interface to which you are applying the Cisco Easy VPN remote configuration.
Step 8	**crypto ipsec client ezvpn** *name* [**outside** | **inside**] **Example:** `Router(config-if)# crypto ipsec client ezvpn ezvpnclient outside`	Assigns the Cisco Easy VPN remote configuration to the WAN interface which causes the router to automatically create the NAT or PAT[1] and the access list configuration needed for the VPN connection.
Step 9	**exit** **Example:** `Router(config-crypto-ezvpn)# exit`	Returns to global configuration mode.

1. PAT = port address translation

Configuration Example

The following configuration example shows the EasyVPN client configuration.

```
!
aaa new-model
!
aaa authentication login rtr-remote local
aaa authorization network rtr-remote local
aaa session-id common
!
username username1 password 0 password1
!
crypto isakmp policy 1
    encryption 3des
    authentication pre-share
    group 2
    lifetime 480
!
crypto isakmp client configuration group rtr-remote
    key secret-password
    dns 10.50.10.1 10.60.10.1
    domain company.com
    pool dynpool
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto ipsec security-association lifetime seconds 86400
!
crypto dynamic-map dynmap 1
    set transform-set vpn1
    reverse-route
!
crypto map static-map 1 ipsec-isakmp dynamic dynmap
crypto map dynmap isakmp authorization list rtr-remote
crypto map dynmap client configuration address respond
```

```
crypto ipsec client ezvpn ezvpnclient
    connect auto
    group 2 key secret-password
    mode client
    peer 192.168.100.1
!

interface gigabitethernet 0/4
    crypto ipsec client ezvpn ezvpnclient outside
    crypto map static-map

interface vlan 1
    crypto ipsec client ezvpn ezvpnclient inside
!
```

Configure a Site-to-Site GRE Tunnel

To configure a site-to-site GRE tunnel, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **interface** *type number*
2. **ip address** *ip-address mask*
3. **tunnel source** *interface-type number*
4. **tunnel destination** *default-gateway-ip-address*
5. **crypto map** *map-name*
6. **exit**
7. **ip access-list** {**standard** | **extended**} *access-list-name*
8. **permit** *protocol source source-wildcard destination destination-wildcard*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:** `Router(config)# interface tunnel 1`	Creates a tunnel interface and enters interface configuration mode.
Step 2	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.62.1.193 255.255.255.252`	Assigns an address to the tunnel.

	Command or Action	Purpose
Step 3	**tunnel source** *interface-type number* **Example:** `Router(config-if)# tunnel source gigabitethernet 0/0`	Specifies the source endpoint of the router for the GRE tunnel.
Step 4	**tunnel destination** *default-gateway-ip-address* **Example:** `Router(config-if)# tunnel destination 192.168.101.1`	Specifies the destination endpoint of the router for the GRE tunnel.
Step 5	**crypto map** *map-name* **Example:** `Router(config-if)# crypto map static-map`	Assigns a crypto map to the tunnel. **Note** Dynamic routing or static routes to the tunnel interface must be configured to establish connectivity between the sites..
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**ip access-list** {**standard** | **extended**} *access-list-name* **Example:** `Router(config)# ip access-list extended vpnstatic1`	Enters ACL[1] configuration mode for the named ACL that the crypto map uses.
Step 8	**permit** *protocol source source-wildcard destination destination-wildcard* **Example:** `Router(config-acl)# permit gre host 192.168.100.1 host 192.168.101.1`	Specifies that only GRE traffic is permitted on the outbound interface.
Step 9	**exit** **Example:** `Router(config-acl)# exit`	Returns to global configuration mode.

1. ACL = access control list

Configuration Example

The following configuration example shows a portion of the configuration file for a site-to-site VPN using a GRE tunnel as described in the preceding sections.

```
!
aaa new-model
!
aaa authentication login rtr-remote local
aaa authorization network rtr-remote local
aaa session-id common
!
username username1 password 0 password1
!
interface tunnel 1
    ip address 10.62.1.193 255.255.255.252

tunnel source GigabitEthernet 0/3

tunnel destination interface 192.168.101.1

ip route 20.20.20.0 255.255.255.0 tunnel 1

crypto isakmp policy 1
    encryption 3des
    authentication pre-share
    group 2
!
crypto isakmp client configuration group rtr-remote
    key secret-password
    dns 10.50.10.1 10.60.10.1
    domain company.com
    pool dynpool
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto ipsec security-association lifetime seconds 86400
!
crypto dynamic-map dynmap 1
    set transform-set vpn1
    reverse-route
!
crypto map static-map 1 ipsec-isakmp dynamic dynmap
crypto map dynmap isakmp authorization list rtr-remote
crypto map dynmap client configuration address respond
!
! Defines the key association and authentication for IPsec tunnel.
crypto isakmp policy 1
hash md5
authentication pre-share
crypto isakmp key cisco123 address 200.1.1.1
!
!
! Defines encryption and transform set for the IPsec tunnel.
crypto ipsec transform-set set1 esp-3des esp-md5-hmac
!
! Associates all crypto values and peering address for the IPsec tunnel.
crypto map to_corporate 1 ipsec-isakmp
 set peer 200.1.1.1
 set transform-set set1
 match address 105
!
!
! VLAN 1 is the internal home network.
interface vlan 1
 ip address 10.1.1.1 255.255.255.0
 ip nat inside
 ip inspect firewall in ! Inspection examines outbound traffic.
    crypto map static-map
```

```
    no cdp enable
!
! GE4 is the outside or Internet-exposed interface
interface Gigabitethernet 0/4
 ip address 210.110.101.21 255.255.255.0
 ! acl 103 permits IPsec traffic from the corp. router as well as
 ! denies Internet-initiated traffic inbound.
 ip access-group 103 in
 ip nat outside
 no cdp enable
 crypto map to_corporate ! Applies the IPsec tunnel to the outside interface.
!
! Utilize NAT overload in order to make best use of the
! single address provided by the ISP.
ip nat inside source list 102 interface Gigabitethernet 0/1 overload
ip classless
ip route 0.0.0.0 0.0.0.0 210.110.101.1
no ip http server
!
!
! acl 102 associated addresses used for NAT.
access-list 102 permit ip 10.1.1.0 0.0.0.255 any
! acl 103 defines traffic allowed from the peer for the IPsec tunnel.
access-list 103 permit udp host 200.1.1.1 any eq isakmp
access-list 103 permit udp host 200.1.1.1 eq isakmp any
access-list 103 permit esp host 200.1.1.1 any
! Allow ICMP for debugging but should be disabled because of security implications.
access-list 103 permit icmp any any
access-list 103 deny ip any any ! Prevents Internet-initiated traffic inbound.
! acl 105 matches addresses for the IPsec tunnel to or from the corporate network.
access-list 105 permit ip 10.1.1.0 0.0.0.255 192.168.0.0 0.0.255.255
no cdp run
```

Configuring Dynamic Multipoint VPN

The Dynamic Multipoint VPN (DMVPN) feature is a simplified solution to deploy large and small IP Security (IPsec) VPNs by combining GRE tunnels, IPsec encryption, and Next Hop Resolution Protocol (NHRP). DMVPN simplifies the configuration tasks in a large scale VPN deployment and reduces the administrative overhead.

DMVPN is useful in a scenario, when one central router at the head office acts as a hub and other branch routers act as spoke and connected to the hub router to access the company's resources. DMVPN is also useful for spoke to spoke deployment and can be used for branch-to-branch interconnections

See the Example: DMVPN Configuration, page 83 for a typical DMVPN configuration for a hub and spoke deployment. For additional information about configuring DMVPN, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_dmvpn/configuration/15-mt/sec-conn-dmvpn-15-mt-book/sec-conn-dmvpn-dmvpn.html

Example: DMVPN Configuration

The following configuration example shows the configuration for DMVPN hub and spoke deployment model. In this example, Cisco 800M series ISR is configured as spoke and Cisco 2900 Series ISR is configured as hub. For readability some part of the configuration is removed.

This configuration section shows the configuration of 800M Series ISR as a spoke.

800M_spoke# **show running-config**

```
Building configuration...
Current configuration : 2546 bytes
!
! Last configuration change at 09:09:39 UTC Tue Jun 24 2014
!
version 15.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 800M_spoke
!
boot-start-marker
boot-end-marker
!
!
logging buffered 10000000
!
no aaa new-model
!
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
crypto isakmp policy 1
 encr aes
 hash sha256
 authentication pre-share
 group 2
crypto isakmp key ISA_KEY address 0.0.0.0
crypto isakmp keepalive 10 periodic
!

crypto ipsec transform-set DMVPN-TRANS-SET esp-aes 256 esp-sha-hmac
 mode tunnel
!
crypto ipsec profile DMVPN-PROFILE
 set security-association lifetime seconds 120
 set transform-set DMVPN-TRANS-SET
!

interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
interface Tunnel0
 ip address 24.1.1.2 255.255.255.0
 no ip redirects
 ip mtu 1440
 ip nhrp authentication ISA_KEY
 ip nhrp map multicast 172.16.0.1
 ip nhrp map 24.1.1.1 172.16.0.1
 ip nhrp network-id 1
 ip nhrp holdtime 120
 ip nhrp nhs 24.1.1.1
 ip nhrp registration timeout 30
 ip nhrp shortcut
 tunnel source GigabitEthernet0/9
 tunnel mode gre multipoint
 tunnel key 0
 tunnel protection ipsec profile DMVPN-PROFILE
!
```

```
interface GigabitEthernet0/0
 no ip address
!
interface GigabitEthernet0/1
 no ip address
!
interface GigabitEthernet0/2
 no ip address
!
interface GigabitEthernet0/3
 no ip address
!
interface GigabitEthernet0/4
 no ip address
!
interface GigabitEthernet0/5
 no ip address
!
interface GigabitEthernet0/6
 no ip address
!
interface GigabitEthernet0/7
 no ip address
!
interface GigabitEthernet0/8
 ip address 192.168.3.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/9
 ip address 172.15.0.1 255.255.255.0
 duplex auto
 speed auto
!
interface Vlan1
 ip address 190.160.10.111 255.255.255.0
!
!
router eigrp 20
 network 2.2.2.0 0.0.0.255
 network 24.1.1.0 0.0.0.255
!
!
router eigrp 10
 network 172.15.0.0 0.0.0.255
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 192.168.4.0 255.255.255.0 100.100.100.2
ip route 192.168.5.0 255.255.255.0 100.100.100.2
!
access-list 101 permit ip 192.168.3.0 0.0.0.255 192.168.4.0 0.0.0.255
access-list 102 permit ip 100.100.100.0 0.0.0.255 200.200.200.0 0.0.0.255
!
control-plane
!
!
line con 0
 no modem enable
line vty 0 4
 login
```

```
 transport input none
!
scheduler allocate 20000 1000
!
end
```

This configuration section shows the configuraton of 2900 Series ISR as hub.

2901_hub# **show running-config**

```
Building configuration...

Current configuration : 3210 bytes
!
! Last configuration change at 07:34:35 UTC Tue Jun 24 2014
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 2901_hub
!
boot-start-marker
boot-end-marker
!
!
logging buffered 10000000
!
no aaa new-model
!
ip cef
!
!
no ipv6 cef
!
multilink bundle-name authenticated
!
license udi pid CISCO2901/K9 sn FGL180322RF
license boot module c2900 technology-package securityk9
!
!
!
redundancy
!

lldp run
!
!
crypto isakmp policy 1
 encr aes
 hash sha256
 authentication pre-share
 group 2
crypto isakmp key ISA_KEY address 0.0.0.0
crypto isakmp keepalive 10 periodic
!
!
crypto ipsec transform-set DMVPN-TRANS-SET esp-aes 256 esp-sha-hmac
 mode tunnel
!
crypto ipsec profile DMVPN-PROFILE
 set security-association lifetime seconds 120
 set transform-set DMVPN-TRANS-SET
!
!

interface Loopback0
 ip address 1.1.1.1 255.255.255.255
 ip ospf message-digest-key 1 md5 cisco
!
```

```
interface Loopback1
 ip address 12.12.12.2 255.255.255.255
!
interface Loopback2
 ip address 12.12.12.3 255.255.255.255
!
interface Loopback3
 ip address 12.12.12.4 255.255.255.255
!
interface Loopback4
 ip address 12.12.12.5 255.255.255.255
!
interface Tunnel0
 ip address 24.1.1.1 255.255.255.0
 no ip redirects
 ip mtu 1440
 no ip split-horizon eigrp 10
 ip nhrp authentication ISA_KEY
 ip nhrp map multicast dynamic
 ip nhrp network-id 1
 ip nhrp shortcut
 ip nhrp redirect
 ip summary-address eigrp 20 192.168.0.0 255.255.0.0
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 0
 tunnel protection ipsec profile DMVPN-PROFILE
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 192.168.5.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 172.16.0.1 255.255.255.0
 ip ospf message-digest-key 1 md5 cisco
 ip ospf priority 10
 duplex auto
 speed auto
!
interface GigabitEthernet0/1/0
 switchport access vlan 2
 no ip address
 shutdown
!
interface GigabitEthernet0/1/1
 switchport access vlan 10
 no ip address
!
interface GigabitEthernet0/1/2
 switchport access vlan 10
 no ip address
!
interface GigabitEthernet0/1/3
 switchport access vlan 20
 no ip address
!
interface GigabitEthernet0/1/4
 no ip address
!
```

```
interface GigabitEthernet0/1/5
 switchport access vlan 10
 no ip address
!
interface GigabitEthernet0/1/6
 no ip address
!
interface GigabitEthernet0/1/7
 no ip address
!
interface Vlan1
 no ip address
!
!
router eigrp 10
 network 172.16.0.0 0.0.0.255
!
!
router eigrp 20
 network 1.1.1.0 0.0.0.255
 network 24.1.1.0 0.0.0.255
 network 192.168.5.0
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 100.100.100.0 255.255.255.0 150.150.150.2
ip route 192.168.3.0 255.255.255.0 150.150.150.2
ip route 192.168.4.0 255.255.255.0 150.150.150.2
ip route 200.200.200.0 255.255.255.0 150.150.150.2
!
!

control-plane
!

line con 0
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Configuring Group Encrypted Transport VPN

Group Encrypted Transport VPN (GETVPN) is a tunnel-less VPN technology that provides end-to-end security for network traffic in a native mode and maintain the mesh topology. GET VPN combines the keying protocol Group Domain of Interpretation (GDOI) with IPsec encryption to provide users with an efficient method of securing IP multicast traffic or unicast traffic. GET VPN enables the router to apply encryption to tunnel-less (native) IP multicast and unicast packets and eliminates the requirement to configure tunnels to protect multicast and unicast traffic.

By removing the need for point-to-point tunnels, meshed networks can scale higher while maintaining network-intelligence features that are critical to voice and video quality, such as QoS, routing, and multicast. GET VPN offers a new standards-based IP security (IPsec) security model that is based on the concept of "trusted" group members. Trusted member routers use a common security methodology that is independent of any point-to-point IPsec tunnel relationship.

A GETVPN deployment has primarily three components, Key Server (KS), Group Member (GM), and Group Domain of Interpretation (GDOI) protocol. GMs encrypt or decrypt the traffic and KS distributes the encryption key to all the group members. The KS decides on one single data encryption key for a given life time. Since all GMs use the same key, any GM can decrypt the traffic encrypted by any other GM. GDOI protocol is used between the GM and KS for group key and group SA management. Minimum one KS is required for a GETVPN deployment.

Unlike traditional IPSec encryption solutions, GET VPN uses the concept of group security association (SA). All members in the GETVPN group can communicate with each other using a common encryption policy and a shared SA and therefore no need to negotiate IPSec between GMs on a peer to peer basis; thereby reducing the resource load on the GM routers.

See the Example: GETVPN Configuration, page 90 for a sample GETVPN deployment configuration.

For additional information about configuring GET VPN, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_getvpn/configuration/15-mt/sec-get-vpn-15-mt-book/sec-get-vpn.html

Example: GETVPN Configuration

The following configuration example shows the configuration for GETVPN deployment. In this example, a Cisco 800M series ISR is configured as GM and the Cisco 1900 Series ISR is configured as KS.

This configuration section shows the configuration of 800M Series ISR as GM.

800M_GM# **show running-config**

```
Building configuration...

Current configuration : 1752 bytes
!
!
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 800M_GM
!
boot-start-marker
boot-end-marker
!
```

```
no aaa new-model
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2
!
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
cts logging verbose
license udi pid C841M-8X/K9 sn FOC18170PNJ
license accept end user agreement
license boot module c800m level advipservices
!
redundancy
!

crypto isakmp policy 100
 encr aes
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key cisco address 192.168.1.2
!
crypto gdoi group gdoi
 identity number 1234
 server address ipv4 192.168.1.2

!
crypto map crypto 10 gdoi
 set group gdoi
!
interface GigabitEthernet0/0
 no ip address
!
interface GigabitEthernet0/1
 no ip address
!
interface GigabitEthernet0/2
 no ip address
!
interface GigabitEthernet0/3
 no ip address
!
interface GigabitEthernet0/4
 no ip address
!
interface GigabitEthernet0/5
 no ip address
!
interface GigabitEthernet0/6
 no ip address
!
interface GigabitEthernet0/7
 no ip address
!
interface GigabitEthernet0/8
 ip address 10.1.3.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/9
 ip address 192.168.3.2 255.255.255.0
```

```
 duplex auto
 speed auto
 crypto map crypto
!
interface Vlan1
 no ip address
!
!
router eigrp 1
 network 10.1.3.0 0.0.0.255
 network 192.168.3.0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
control-plane
!
line con 0
 no modem enable
line vty 0 4
 login
 transport input none
!
scheduler allocate 20000 1000
!
end
```

This configuration section shows the configuration of Cisco 1900 Series ISR as KS.

1921_KS# **show running-config**

```
Building configuration...
Current configuration : 2019 bytes
!
version 15.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 1921_KS
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
!

!
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!

license udi pid CISCO1921/K9 sn FGL155022DY
license boot module c1900 technology-package securityk9
license boot module c1900 technology-package datak9
!
!
!
```

```
redundancy
!
crypto isakmp policy 100
 encr aes
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key cisco address 0.0.0.0
!

crypto ipsec transform-set trans esp-aes esp-sha-hmac
 mode tunnel
!
!
crypto ipsec profile ipsec
 set transform-set trans
!
crypto gdoi group gdoi
 identity number 1234
 server local
  rekey algorithm aes 256
  rekey lifetime seconds 3600
  rekey authentication mypubkey rsa vpnkeys
  rekey transport unicast
  sa ipsec 10
   profile ipsec
   match address ipv4 getvpn
   replay counter window-size 64
   no tag
  address ipv4 192.168.1.2
!
!
crypto map crypto 10 gdoi
 set group gdoi
!

interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 192.168.1.2 255.255.255.0
 duplex auto
 speed auto
 crypto map crypto
!
interface Serial0/0/0
 no ip address
 shutdown
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!

router eigrp 1
 network 192.168.1.0
```

```
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!

ip access-list extended getvpn
 permit ip 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
!

control-plane
!

line con 0
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport output lat pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Configuring SSL VPN

The Secure Socket Layer Virtual Private Network (SSL VPN) feature provides support for remote user access to enterprise networks from anywhere on the Internet. Remote access is provided through a SSL–enabled SSL VPN gateway. The SSL VPN gateway allows remote users to establish a secure VPN tunnel using a web browser. This feature provides a comprehensive solution that allows easy access to a broad range of web resources and web-enabled applications using native HTTP over SSL (HTTPS) browser support. SSL VPN delivers three modes of SSL VPN access: clientless, thin-client, and full-tunnel client support.

See the "Example: SSL VPN Configuration" section for a sample SSL VPN gateway configuration.

For additional information about configuring SSL VPN, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_sslvpn/configuration/15-mt/sec-conn-sslvpn-15-mt-book/sec-conn-sslvpn-ssl-vpn.html

Example: SSL VPN Configuration

This configuration example shows the configuration for SSL VPN gateway using Cisco 800M Series ISR.

800M# **show running-config**

```
Building configuration...

Current configuration : 4053 bytes
!
version 15.5
```

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 800M
!
boot-start-marker
boot-end-marker
!
!
!
aaa new-model
!
!
aaa authentication login default local
aaa authentication login ciscocp_vpn_xauth_ml_1 local
!
!

aaa session-id commont
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2
!
crypto pki trustpoint TP-self-signed-2716339910
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2716339910
 revocation-check none
 rsakeypair TP-self-signed-2716339910
!
!
crypto pki certificate chain TP-self-signed-2716339910
 certificate self-signed 01
  3082022B 30820194 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32373136 33333939 3130301E 170D3134 31313132 31313430
  35355A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 37313633
  33393931 3030819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
  8100A775 D34D41D6 281317C5 427BBC6D 3D97F5B4 F91E924B AB23F5CC F92336E6
  29EBDC57 45A455B7 D7300C0C 07C5DDF8 62E2BDFB CDEB57CC EFAE7006 A72D4C20
  2D9995E7 472D2C4E 079828B3 B63DDB66 A9D3D77F BC844CBD 255D81F0 84564748
  4FAD69E1 94F5AFC9 0450EFDC 9096BD38 3F4FA022 0680E969 174197EA 3F85DD4C
  B1490203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF 301F0603
  551D2304 18301680 145602C5 80924574 A895C527 F177A81B 4EA03C94 EA301D06
  03551D0E 04160414 5602C580 924574A8 95C527F1 77A81B4E A03C94EA 300D0609
  2A864886 F70D0101 05050003 81810090 823846F0 FAA084FB F5C17F04 00E11E54
  D9D9B32A 4EBB96D4 8414C5DD 0DB8728B 84518031 0B22A20A 989C341C 4AB15B7B
  B192E99B E29138E9 56263016 5565DEAA 9CE9E40B D945EF2C 1BFE110C 4622F707
  39E7FA48 DA3B15DD CA66AA8F 61783562 7C09932F BD4E5AB4 A1242A71 90E27B22
  71CD3A0D A0004521 D1DB1E2C D95BEF
        quit
!

ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
cts logging verbose
license udi pid C841M-8X/K9 sn FCW1842005Y
!
!
username cisco privilege 15 password 0 cisco
!
```

```
redundancy
!
crypto vpn anyconnect sdflash:/webvpn/anyconnect-win-3.1.03103-k9.pkg sequence 1


!
interface Loopback10
 ip address 100.100.100.100 255.255.255.255
!
interface GigabitEthernet0/0
 no ip address
!
interface GigabitEthernet0/1
 no ip address
!
interface GigabitEthernet0/2
 no ip address
!
interface GigabitEthernet0/3
 no ip address
!
interface GigabitEthernet0/4
 no ip address
!
interface GigabitEthernet0/5
 no ip address
!
interface GigabitEthernet0/6
 no ip address
!
interface GigabitEthernet0/7
 no ip address
!
interface GigabitEthernet0/8
 ip address 192.168.10.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/9
 ip address 9.43.17.81 255.255.0.0
 duplex auto
 speed auto
!
interface Virtual-Template1
 ip unnumbered GigabitEthernet0/8
 ip virtual-reassembly in
!
interface Vlan1
 no ip address
!
ip local pool IP_Pool 10.10.10.1 10.10.10.10
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
ip route 202.153.144.0 255.255.255.0 9.43.0.1
!

control-plane
!

line con 0
 no modem enable
```

```
line vty 0 4
 transport input none
!
scheduler allocate 20000 1000
!

webvpn gateway gateway_1
 ip address 192.168.10.1 port 443
 ssl trustpoint TP-self-signed-2716339910
 inservice
 !
webvpn context Test
 secondary-color white
 title-color #FF9900
 text-color black
 virtual-template 1
 aaa authentication list ciscocp_vpn_xauth_ml_1
 gateway gateway_1
 !
 ssl authenticate verify all
 inservice
 !
 policy group policy_1
   functions svc-enabled
   svc address-pool "IP_Pool" netmask 255.255.255.255
   svc default-domain "cisco.com"
   svc keep-client-installed
   svc rekey time 240
   svc dns-server primary 10.105.130.1
   svc wins-server primary 10.105.130.1
 default-group-policy policy_1
!
end
```

Configuring FlexVPN

FlexVPN is Cisco's implementation of the IKEv2 standard featuring a unified paradigm and CLI that combines site to site, remote access, hub and spoke topologies and partial meshes (spoke to spoke direct). FlexVPN offers a simple but modular framework that extensively uses the tunnel interface paradigm while remaining compatible with legacy VPN implementations using crypto maps.

See the "Example: FlexVPN Configuration" section for a sample FlexVPN hub and spoke configuration.

For additional information about configuring FlexVPN, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_ike2vpn/configuration/15-mt/sec-flex-vpn-15-mt-book/sec-intro-ikev2-flex.html

Example: FlexVPN Configuration

The following configuration example shows the configuration for FlexVPN hub and spoke deployment model. In this example, Cisco 800M series ISR is configured as a spoke and Cisco 3900 Series ISR is configured as the hub.

This configuration section shows the configuration of 800M Series ISR as a spoke.

800M# **show running-config**

```
Building configuration...
```

```
Current configuration : 2461 bytes
!
!
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 800M
!
boot-start-marker
boot-end-marker
!

aaa new-model
!
!
aaa authorization network FLEX local
!

aaa session-id common
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2

!
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
chat-script multimode "" "AT!CALL" TIMEOUT 20 "OK"
cts logging verbose
license udi pid C841M-4X/K9 sn FCW1839001E
!

redundancy
!
crypto ikev2 authorization policy FLEX
 route set interface
!
!
!
crypto ikev2 keyring KEYRING
 peer R1
  address 172.16.0.1
  pre-shared-key CISCO
 !
!
!
crypto ikev2 profile default
 match identity remote address 172.16.0.1 255.255.255.255
 identity local key-id FLEX
 authentication remote pre-share
 authentication local pre-share
 keyring local KEYRING
 aaa authorization group psk list FLEX FLEX
!
!
!
controller Cellular 0/0
 modem link-recovery rssi onset-threshold -110
 modem link-recovery monitor-timer 20
 modem link-recovery wait-timer 10
```

```
 modem link-recovery debounce-count 6

!
interface Loopback0
 ip address 2.2.2.2 255.255.255.0
!
interface Tunnel0
 ip address negotiated
 tunnel source GigabitEthernet0/5
 tunnel mode ipsec ipv4
 tunnel destination 172.16.0.1
 tunnel protection ipsec profile default
!
interface GigabitEthernet0/0
 no ip address
!
interface GigabitEthernet0/1
 no ip address
!
interface GigabitEthernet0/2
 no ip address
!
interface GigabitEthernet0/3
 no ip address
!
interface GigabitEthernet0/4
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/5
 ip address 172.16.0.2 255.255.255.0
 duplex auto
 speed auto
!
interface Cellular0/0/0
 no ip address
 encapsulation slip
 dialer in-band
 dialer string multimode
!
interface Serial0/1/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Vlan1
 no ip address
!
!
router eigrp 1
 network 0.0.0.0
 passive-interface default
 no passive-interface Tunnel0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!

control-plane
!
```

```
line con 0
 no modem enable
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 stopbits 1
line 3
 script dialer multimode
 no exec
line vty 0 4
 transport input none
!
scheduler allocate 20000 1000
!
end
```

This configuration section shows the configuration of 800M Series ISR as a spoke.

C3900# **show running-config**

```
Building configuration...

Current configuration : 2690 bytes
!
! Last configuration change at 13:10:19 UTC Fri Oct 31 2014
version 15.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname C3900
!
boot-start-marker
boot-end-marker
!
aqm-register-fnf
!
!
aaa new-model
!
!
aaa authorization network LOCALIKEv2 local

!
!
aaa session-id common

!
!
!
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!

!
voice-card 0
!

!
```

```
license udi pid C3900-SPE200/K9 sn FOC16075NAN
license accept end user agreement
license boot module c3900e technology-package securityk9
license boot module c3900e technology-package datak9
!
!
!
redundancy
!
crypto ikev2 authorization policy AUTHOR-POLICY
 pool POOL
!
!
!
crypto ikev2 keyring KEYRING
 peer R2
  address 172.16.0.2
  pre-shared-key CISCO
 !
!
!
crypto ikev2 profile default
 match identity remote key-id FLEX
 authentication remote pre-share
 authentication local pre-share
 keyring local KEYRING
 aaa authorization group psk list LOCALIKEv2 AUTHOR-POLICY
 virtual-template 1

!
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.0
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 172.16.0.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1/0
 no ip address
!
interface FastEthernet0/1/1
 no ip address
!
interface FastEthernet0/1/2
 no ip address
```

```
!
interface FastEthernet0/1/3
 no ip address
!
interface FastEthernet0/1/4
 no ip address
!
interface FastEthernet0/1/5
 no ip address
!
interface FastEthernet0/1/6
 no ip address
!
interface FastEthernet0/1/7
 no ip address
!
interface FastEthernet0/1/8
 no ip address
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
 tunnel source GigabitEthernet0/1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile default
!
interface Vlan1
 no ip address
!
!
!
router eigrp 1
 network 1.1.1.1 0.0.0.0
 passive-interface default
 no passive-interface Virtual-Template1
!
ip local pool POOL 192.168.0.1 192.168.0.10
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
!
!
nls resp-timeout 1
cpd cr-id 1
!
!
!
!
control-plane
!
 !

mgcp behavior rsip-range tgcp-only
mgcp behavior comedia-role none
mgcp behavior comedia-check-media-src disable
mgcp behavior comedia-sdp-force disable
!
mgcp profile default
!

gatekeeper
 shutdown
!
```

```
!
!
line con 0
line aux 0
line vty 0 4
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Configuring Zone-Based Policy Firewall

Zone-Based Policy Firewall (also known as Zone-Policy Firewall, or ZFW) changes the firewall configuration from the interface-based model to a more flexible, more easily understood zone-based model. Interfaces are assigned to zones, and inspection policy is applied to traffic moving between the zones. Inter-zone policies offer considerable flexibility and granularity, so different inspection policies can be applied to multiple host groups connected to the same router interface.

For more information about configuring zone-based policy firewall, see the following weblink:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_zbf/configuration/15-mt/sec-data-zbf-15-mt-book/sec-zone-pol-fw.html

Configuring VRF-Aware Cisco Firewall

VRF-Aware Cisco Firewall applies Cisco Firewall functionality to Virtual Routing and Forwarding (VRF) interfaces when the firewall is configured on a service provider (SP) or large enterprise edge device. SPs can provide managed services to small and medium business markets.

For more information about configuring VRF-aware Cisco Firewall, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_zbf/configuration/15-mt/sec-data-zbf-15-mt-book/sec-cbac-vrf-fw.html

Configuring Subscription-Based Cisco IOS Content Filtering

The Subscription-based Cisco IOS Content Filtering feature interacts with the Trend Micro URL filtering service so that HTTP requests can be allowed or blocked, and logged, based on a content filtering policy. The content filtering policy specifies how to handle items such as web categories, reputations (or security ratings), trusted domains, untrusted domains, and keywords. URLs are cached on the router, so that subsequent requests for the same URL do not require a lookup request, thus improving performance.

For more information about configuring subscription-based Cisco IOS content filtering see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_zbf/configuration/15-mt/sec-data-zbf-15-mt-book/subscrip-cont-filter.html

Configuring On-Device Management for Security Features

The On-Device Management for Security Features provides an intuitive and simple management interface, the Cisco Configuration Professional Express, to deploy a variety of security features. You can deploy security features including zone-based firewalls, VPN, Intrusion Detection System (IDS) and URL filtering through the Cisco Configuration Professional Express.

The Cisco Configuration Professional Express uses existing zone-based firewall CLIs in conjunction with Network-Based Application Recognition 2 (NBAR2) CLIs to determine the application category, and position NBAR2 protocols supported by the firewall into the relevant application category.

Fro more information about enabling NBAR2 for zone-based firewalls, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_zbf/configuration/15-mt/sec-data-zbf-15-mt-book/on-device-mgmt.html

Related Documents

Topic	Document Title
DMVPN	Dynamic Multipoint VPN Configuration Guide, Cisco IOS Release 15M&T
GETVPN	Cisco Group Encrypted Transport VPN Configuration Guide, Cisco IOS Release 15M&T
SSL VPN	SSL VPN Configuration Guide, Cisco IOS Release 15M&T
FlexVPN	FlexVPN and Internet Key Exchange Version 2 Configuration Guide, Cisco IOS Release 15M&T
IKE for IPSec VPNs	Internet Key Exchange for IPsec VPNs Configuration Guide, Cisco IOS Release 15M&T

Configuring QoS

This chapter provides information about configuring the Quality of Service (QoS) features on the Cisco 800M Series ISR and contains the following sections:

Configuring Class Based Weighted Fair Queuing

Class Based Weighted Fair Queuing (CBWFQ) provides congestion-management support for user-defined traffic classes. For CBWFQ, you define traffic classes based on match criteria including protocols, access control lists (ACLs), and input interfaces. Packets satisfying the match criteria for a class constitute the traffic for that class. A FIFO queue is reserved for each class, and traffic belonging to a class is directed to the queue for that class.

Once a class has been defined according to its match criteria, you can assign it characteristics. To characterize a class, you assign it bandwidth, weight, and maximum packet limit. The bandwidth assigned to a class is the guaranteed bandwidth delivered to the class during congestion.

For more information about configuring CBWFQ see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_conmgt/configuration/15-mt/qos-conmgt-15-mt-book/qos-conmgt-cfg-wfq.html

Example: Class Based Weighted Fair Queuing

In this example, two class maps are created and their match criteria are defined. For the first class map called class1, the numbered ACL 101 is used as the match criterion. For the second map class, called class2, the numbered ACL 102 is used as the match criterion. Packets are checked against the contents of these ACLs to determine if they belong to the class.

```
Router# configure terminal
Router(config)# access-list 101 permit udp host 10.10.10.10 host 10.10.10.20 range 16384
20000
Router(config# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000
56000
Router(config)# class-map class1
Router(config-cmap)# match access-group 101
Router(config-cmap)# exit
Router(config-cmap)# class-map class2
Router(config-cmap)# match access-group 102
Router(config-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 3000
Router(config-pmap-c)# queue-limit 30
Router(config-pmap-c)# exit
Router(config-pmap)# class class2
Router(config-pmap-c)# bandwidth 2000
Router(config)# interface gigabitethernet 0/4
Router(config-if)# service output policy1
Router(config-if)# exit
```

Configuring Low-Latency Queueing

Strict priority queueing allows delay-sensitive data such as voice to be dequeued and sent before packets in other queues are dequeued. Low Latency Queuing (LLQ) provides strict priority queueing for CBWFQ, reducing jitter in voice conversations. LLQ enables use of a single, strict priority queue within CBWFQ at the class level, allowing you to direct traffic belonging to a class to the CBWFQ strict priority queue. To enqueue class traffic to the strict priority queue, you specify the named class within a policy map and then configure the priority command for the class. Within a policy map, you can give priority status to one or more classes. When multiple classes within a single policy map are configured as priority classes, all traffic from these classes is enqueued to the same, single, strict priority queue.

For more information on configuring low latency queuing see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_conmgt/configuration/15-mt/qos-conmgt-15-mt-book/qos-conmgt-cfg-wfq.html

Example: Low-Latency Queueing

```
Router# configure terminal
Router(config)# class-map voice
Router(config-cmap)# match access-group 102
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50 60
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth 20
Router(config-pmap)# class class-default
```

```
Router(config-pmap-c)# fair-queue
Router(config)# interface serial 0/0/0
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

Configuring Class-Based Traffic Shaping

Traffic shaping allows you to control the traffic going out an interface in order to match its flow to the speed of the remote target interface and to ensure that the traffic conforms to policies contracted for it. Thus, traffic adhering to a particular profile can be shaped to meet downstream requirements, thereby eliminating bottlenecks in topologies with data-rate mismatches.

For more information on class-based traffic shaping, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/configuration/guide/fqos_c/qcfcbshp.html

Example: Class-Based Traffic Shaping

The following example defines a class c1 which is configured to shape traffic to 384 kbps, with a normal burst size of 15440 bits.

```
Router# configure terminal
```

```
Router(config)# policy-map shape
Router(config-pmap)# class c1
Router(config-pmap-c)# shape average 384000 15440
Router(config-pmap-c)# end
Router(config)# interface Serial 0/0/0
Router(config-if)# service out shape
```

Configuring Class-Based Traffic Policing

Class-based trafic policing allows you to control the maximum rate of traffic transmitted or received on an interface. Class-based traffic policing is often configured on interfaces at the edge of a network to limit traffic into or out of the network. In most class-based policing configurations, traffic that falls within the rate parameters is transmitted, whereas traffic that exceeds the parameters is dropped or transmitted with a different priority.

For more information on configuring class-based traffic policing, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_plcshp/configuration/15-mt/qos-plcshp-15-mt-book/qos-plcshp-class-plc.html

Example: Class-Based Traffic Policing

In this example, Class-Based Policing is configured with the average rate at 8000 bits per second and the normal burst size at 1000 bytes for all packets leaving Gigabit Ethernet interface 0/4.

```
Router# configure terminal
Router(config)# class-map access-match
Router(config-cmap)# match access-group 1
Router(config)# policy-map police-setting
Router(config-pmap)# class access-match
```

```
Router(config-pmap-c)# police 8000 1000 1000 conform-action transmit exceed-action drop
Router(config-pmap-c)# violate-action drop
Router(config)# interface gigabitethernet 0/4
Router(config-if)# service-policy output policy-setting
Router(config-if)# exit
```

Configuring Class-Based Weighted Random Early Detection

Weighted Random Early Detection (WRED) combines the capabilities of the Random Early Detection (RED), algorithm with the IP Precedence feature to provide for preferential traffic handling of higher priority packets. WRED can selectively discard lower priority traffic when the interface begins to get congested and provide differentiated performance characteristics for different classes of service.

You can configure WRED to ignore IP precedence when making drop decisions so that nonweighted RED behavior is achieved.WRED makes early detection of congestion possible and provides for multiple classes of traffic. It also protects against global synchronization. For these reasons, WRED is useful on any output interface where you expect congestion to occur.

For more information about configuring WRED, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_conavd/configuration/15-mt/qos-conavd-15-mt-book/qos-conavd-cfg-wred.html

Example: Class-Based Weighted Random Early Detection

```
Router(config-if)# class-map c1
Router(config-cmap)# match access-group 101
Router(config-if)# policy-map p1
Router(config-pmap)# class c1
Router(config-pmap-c)# bandwidth 48
Router(config-pmap-c)# random-detect dscp-based
Router(config-pmap-c)# random-detect dscp 8 24 40
Router(config-if)# service-policy output p1
```

Configuring QoS Hierarchical Queueing Framework

The QoS Hierarchical Queueing Framework (HQF) feature enables you to manage quality of service (QoS) at three different levels: the physical interface level, the logical interface level, and the class level for QoS queueing and shaping mechanisms by using the modular QoS command-line interface (MQC) to provide a granular and flexible overall QoS architecture.

For more information about configuring hierarchical queueing framework see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_hrhqf/configuration/15-mt/qos-hrhqf-15-mt-book/qos-hrhqf.html

Configuring Network-Based Application Recognition

Network-Based Application Recognition (NBAR) is a classification engine that recognizes and classifies a wide variety of protocols and applications. When NBAR recognizes and classifies a protocol or application, the network can be configured to apply the appropriate quality of service (QoS) for that

application or traffic with that protocol.

For more information about configuring NBAR, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_nbar/configuration/15-mt/qos-nbar-15-mt-book.html

Example: Network Based Application Recognition

```
Router# configure terminal
Router(config)# class-map cmap1
Router(config-cmap)# match protocol citrix
Router(config-cmap)# end
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Device(config-pmap-c)# bandwidth percent 50
Device(config-pmap-c)# end
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 0/4
Device(config-if)# service-policy input policy1
Device(config-if)# end
```

Configuring Resource Reservation Protocol

Resource Reservation Protocol (RSVP) is the industry-standard protocol for dynamically setting up end-to-end QoS across a heterogeneous network. RSVP, which runs over IP, allows an application to dynamically reserve network bandwidth. Using RSVP, applications can request a certain level of QoS for a data flow across a network.

The Cisco IOS QoS implementation allows RSVP to be initiated within the network using configured proxy RSVP. Using this capability, you can take advantage of the benefits of RSVP in the network even for non-RSVP enabled applications and hosts. RSVP is designed to guarantee network bandwidth from end-to-end for IP networks.

For more information about configuring RSVP, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_rsvp/configuration/15-mt/qos-rsvp-15-mt-book/config-rsvp.html

Configuring Quality of Service for VPNs

The QoS for VPNs feature provides a solution for making Cisco IOS QoS services operate in conjunction with tunneling and encryption on an interface. Cisco IOS software can classify packets and apply the appropriate QoS service before the data is encrypted and tunneled. The QoS for VPN feature allows users to look inside the packet so that packet classification can be done based on original port numbers and based on source and destination IP addresses. This allows the service provider to treat mission critical or multi-service traffic with higher priority across their network.

For more information about configuring QoS for VPNs see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_classn/configuration/15-mt/qos-classn-15-mt-book/qos-classn-vpn.html

Configuring Per Tunnel QoS for DMVPN

The Per-Tunnel QoS for DMVPN feature lets you apply a quality of service (QoS) policy on a Dynamic Multipoint VPN (DMVPN) hub on a per-tunnel instance (per-spoke basis) in the egress direction for DMVPN hub-to-spoke tunnels. The QoS policy on a DMVPN hub on a per-tunnel instance lets you shape tunnel traffic to individual spokes (a parent policy) and differentiate individual data flows going through the tunnel for policing (a child policy). The QoS policy that the hub uses for a specific spoke is selected according to the specific Next Hop Resolution Protocol (NHRP) group into which that spoke is configured. Although you can configure many spokes into the same NHRP group, the tunnel traffic for each spoke is measured individually for shaping and policing.You can use this feature with DMVPN with or without Internet Protocol Security (IPsec).

For more information about configuring Per-Tunnel QoS for DMVPN, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_dmvpn/configuration/15-mt/sec-conn-dmvpn-15-mt-book/sec-conn-dmvpn-per-tunnel-qos.html

Configuring Layer 2 Auto QoS

You can use the auto-QoS feature to simplify the deployment of QoS features. Auto-QoS determines the network design and enables QoS configurations so that the router can prioritize different traffic flows. It uses the ingress and egress queues instead of using the default (disabled) QoS behavior. The switch offers best-effort service to each packet, regardless of the packet contents or size, and sends it from a single queue.

When you enable auto-QoS, it automatically classifies traffic based on the traffic type and ingress packet label. The switch uses the classification results to choose the appropriate egress queue.

For more information about configuring Auto QoS, see the following link:

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750/software/release/12-2_55_se/configuration/guide/scg3750/swqos.html#wp1231112

Configuring Network Management Features

This chapter provides information about configuring the network management features for the Cisco 800M Series ISR and contains the following sections:

Cisco Configuration Professional

Cisco Configuration Professional is a GUI based device management tool for Cisco access routers. This tool simplifies routing, firewall, IPS, VPN, unified communications, and WAN, and LAN configurations through GUI-based wizards. Cisco CP is a valuable productivity enhancing tool for network administrators and channel partners for deploying routers with increased confidence and ease. It offers a one-click router lock-down and an innovative voice and security auditing capability to check and recommend changes to router configurations. Cisco CP also monitors router status and troubleshoots WAN and VPN connectivity issues.

For more information about configuring Cisco 800M series ISR using Cisco Configuration Professional, see the following web link:

http://www.cisco.com/c/dam/en/us/td/docs/net_mgmt/cisco_configuration_professional/v2_5/olh/ccp.pdf

Cisco Configuration Professional Express

Cisco Configuration Professional Express (Cisco CP Express), a lightweight version of Cisco Configuration Professional, is an embedded, device-management tool that provides the ability to bootstrap and provision a Cisco Integrated Services Router (ISR). The Cisco CP Express helps you set up a network with complete WAN and LAN configuration, along with security features.

For more information about configuring Cisco 800M series ISR using Cisco CP Express, see the following web link:

http://www.cisco.com/c/en/us/td/docs/net_mgmt/cisco_configuration_professional_express/v3_1/guides/featureguide/ccp_express_Feature_Guide.html

Cisco Prime Infrastructure

Cisco Prime Infrastructure is a network management tool that supports life cycle management of your entire network infrastructure from one graphical interface. Prime Infrastructure provides network administrators with a single solution for provisioning, monitoring, optimizing, and troubleshooting both wired and wireless devices. Robust graphical interfaces make device deployments and operations simple and cost-effective.

For more information on configuring Cisco 800M Series ISR using Cisco Prime Infrastructure, see the the following web link:

http://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/infrastructure/2-1/user/guide/pi_ug.html

Embedded Event Manager

Embedded Event Manager (EEM) is a distributed and customized approach to event detection and recovery offered directly in a Cisco IOS device. EEM offers the ability to monitor events and take informational, corrective, or any desired EEM action when the monitored events occur or when a threshold is reached. An EEM policy is an entity that defines an event and the actions to be taken when that event occurs.

For more information on configuring Embedded Event Manager, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/eem/configuration/15-mt/eem-15-mt-book.html

Configuring IP SLAs

IP Service Level Agreements (IP SLAs) allows Cisco customers to analyze IP service levels for IP applications and services, to increase productivity, to lower operational costs, and to reduce the frequency of network outages. IP SLAs uses active traffic monitoring--the generation of traffic in a continuous, reliable, and predictable manner--for measuring network performance. Using IP SLAs, service provider customers can measure and provide service level agreements, and enterprise customers can verify service levels, verify outsourced service level agreements, and understand network performance. IP SLAs can perform network assessments, verify quality of service (QoS), ease the deployment of new services, and assist administrators with network troubleshooting. IP SLAs can be

accessed using the Cisco software commands or Simple Network Management Protocol (SNMP) through the Cisco Round-Trip Time Monitor (RTTMON) and syslog Management Information Bases (MIBs).

For more information on configuring IP SLAs, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_plcshp/configuration/15-mt/qos-plcshp-15-mt-book/qos-plcshp-class-plc.html

Configuring Radius

The RADIUS security system is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco devices and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

For more information about configuring Radius, see the following web link

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_rad/configuration/15-mt/sec-usr-rad-15-mt-book/sec-cfg-radius.html

Configuring TACACS+

TACACS+ is a security application that provides centralized validation of users attempting to gain access to a device or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation.

TACACS+ provides for separate and modular authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service--authentication, authorization, and accounting--independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon. The goal of TACACS+ is to provide a methodology for managing multiple network access points from a single management service.

For more information about configuring TACACS+, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_tacacs/configuration/15-mt/sec-usr-tacacs-15-mt-book/sec-cfg-tacacs.html

Configuring SSH

Secure Shell (SSH) runs on top of a reliable transport layer and provides strong authentication and encryption capabilities.SSH provides a means to securely access and securely execute commands on another computer over a network.

For more information about configuring SSH see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_ssh/configuration/15-mt/sec-usr-ssh-15-mt-book/sec-usr-ssh-sec-shell.html

Configuring SNMP

Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language that is used for monitoring and managing devices in a network.

For more information about configuring SNMP, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/snmp/configuration/15-mt/snmp-15-mt-book/nm-snmp-cfg-snmp-support.html

Configuring NetFlow

NetFlow is a Cisco IOS application that provides statistics on packets flowing through the routing devices in the network. It is emerging as a primary network accounting and security technology.

NetFlow identifies packet flows for both ingress and egress IP packets. It does not involve any connection-setup protocol, either between routers or to any other networking device or end station. NetFlow does not require any change externally--either to the packets themselves or to any networking device. NetFlow is completely transparent to the existing network, including end stations and application software and network devices like LAN switches. Also, NetFlow capture and export are performed independently on each internetworking device; NetFlow need not be operational on each router in the network.

For more information about configuring NetFlow, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/netflow/configuration/15-mt/nf-15-mt-book.html

Configuring Flexible NetFlow

Flexible NetFlow is a Cisco IOS technology that provides statistics on packets flowing through the router. NetFlow is the standard for acquiring IP operational data from IP networks. NetFlow provides data to enable network and security monitoring, network planning, traffic analysis, and IP accounting.

Flexible NetFlow improves on original NetFlow by adding the capability to customize the traffic analysis parameters for your specific requirements. Flexible NetFlow facilitates the creation of more complex configurations for traffic analysis and data export through the use of reusable configuration components.

For more information about configuring NetFlow, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/netflow/configuration/15-mt/nf-15-mt-book.html

MIB Support

The Cisco 800M series ISR supports the MIBs supported by Cisco 800 series ISRs. The following MIBs are modified for Cisco 800M series ISR:

- CISCO-PRODUCTS-MIB
- OLD-CISCO-CHASSIS-MIB
- ENTITY-MIB
- IF-MIB

- CISCO-IF-EXTENSION-MIB
- CISCO-LICENSE-MGMT-MIB
- CISCO-WAN-3G-MIB
- CISCO-ENVMON-MIB
- CISCO-FLASH-MIB

Configuring IP Addressing and IP Services Features

This chapter provides information about configuring IP addressing and IP services features for the Cisco 800M Series ISR and contains the following sections:

Configuring DHCP

Dynamic Host Configuration Protocol (DHCP) is based on the Bootstrap Protocol (BOOTP), which provides the framework for passing configuration information to hosts on a TCP/IP network. DHCP adds the capability to automatically allocate reusable network addresses and configuration options 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, where designated DHCP server hosts allocate network addresses and deliver configuration parameters to dynamically configured hosts. DHCP provides a framework for passing configuration information dynamically to hosts on a TCP/IP network. A DHCP client is an Internet host that uses DHCP to obtain configuration parameters such as an IP address.A DHCP relay agent is any host that forwards DHCP packets between clients and servers. Relay agents are used to forward requests and replies between clients and servers when they are not on the same physical subnet.

For more information on configuring DHCP, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_dhcp/configuration/15-mt/dhcp-15-mt-book.html

Configuring DNS

The Domain Name System (DNS) is a distributed database in which you can map host names to IP addresses through the DNS protocol from a DNS server. Each unique IP address can have an associated hostname. The Cisco IOS software maintains a cache of hostname-to-address mappings for use by the connect, telnet, and ping EXEC commands, and related Telnet support operations. This cache speeds the process of converting names to addresses.

For more information about configuring DNS, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_dns/configuration/15-mt/dns-15-mt-book/dns-config-dns.html

Configuring NAT

Network Address Translation (NAT) enables private IP inter networks that use nonregistered IP addresses to connect to the Internet. NAT operates on a device, usually connecting two networks, and translates the private (not globally unique) addresses in the internal network into legal addresses before packets are forwarded onto another network. NAT can be configured to advertise to the outside world only one address for the entire network. This ability provides additional security by effectively hiding the entire internal network behind that one address.NAT is also used at the enterprise edge to allow internal users access to the Internet and to allow Internet access to internal devices such as mail servers

For more information on configuring NAT, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_nat/configuration/15-mt/nat-15-mt-book/iadnat-addr-consv.html

Configuring NHRP

Next Hop Resolution Protocol (NHRP) is an Address Resolution Protocol (ARP)-like protocol that dynamically maps a non-broadcast multiaccess (NBMA) network. With NHRP, systems attached to an NBMA network can dynamically learn the NBMA (physical) address of the other systems that are part of that network, allowing these systems to directly communicate.

NHRP is a client and server protocol where the hub is the Next Hop Server (NHS) and the spokes are the Next Hop Clients (NHCs). The hub maintains an NHRP database of the public interface addresses of each spoke. Each spoke registers its real address when it boots and queries the NHRP database for real addresses of the destination spokes to build direct tunnels.

For more information on configuring NHRP, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_plcshp/configuration/15-mt/qos-plcshp-15-mt-book/qos-plcshp-class-plc.html

Configuring RIP

Routing Information Protocol (RIP) is a commonly used routing protocol in small to medium TCP/IP networks. It is a stable protocol that uses a distance-vector algorithm to calculate routes.

For more information on configuring RIP, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_plcshp/configuration/15-mt/qos-plcshp-15-mt-book/qos-plcshp-class-plc.html

Configuring EIGRP

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

For more information about configuring EIGRP, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_eigrp/configuration/15-mt/ire-15-mt-book/ire-enhanced-igrp.html

Configuring OSPF

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

For more information about configuring OSPF, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_tacacs/configuration/15-mt/sec-usr-tacacs-15-mt-book/sec-cfg-tacacs.html

Configuring BGP

Border Gateway Protocol (BGP) is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems). The Cisco software implementation of BGP version 4 includes support for 4-byte autonomous system numbers and multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families including IP Version 4 (IPv4), IP Version 6 (IPv6), Virtual Private Networks Version 4 (VPNv4), Connectionless Network Services (CLNS), and Layer 2 VPN (L2VPN).

For more information about configuring BGP, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_ssh/configuration/15-mt/sec-usr-ssh-15-mt-book/sec-usr-ssh-sec-shell.html

Configuring Performance Routing v3

Performance Routing v3 (PfRv3) delivers a set of solutions on automatic prefix and Service Level Agreement (SLA) discovery through an intelligent framework. It provides easier application performance management controls including path optimization, managing over-subscription intelligently in the network for P2P, multi-site deployments, optimizing network infrastructure usage, policy distribution and enforcement and network based bandwidth management.

PfRv3 is an intelligent path control for improving application delivery and WAN efficiency. PfRv3 protects critical application and increases bandwidth utilization and servers as an integral part of the overall Cisco Intelligent WAN (IWAN) solution.

For more information about configuring PfRv3, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/pfrv3/configuration/15-mt/pfrv3-15-mt-book/pfrv3.html

Configuring IP Multicast

IP multicast is a bandwidth-conserving technology that reduces traffic by delivering a single stream of information simultaneously to potentially thousands of businesses and homes. Applications that take advantage of multicast include video conferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news.IP multicast routing enables a host (source) to send packets to a group of hosts (receivers) anywhere within the IP network by using a special form of IP address called the IP multicast group address. The sending host inserts the multicast group address into the IP destination address field of the packet and IP multicast routers and multilayer switches forward incoming IP multicast packets out all interfaces that lead to the members of the multicast group. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message.

For more information about configuring PfRv3, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipmulti_pim/configuration/imc-pim-15-mt-book.html

Configuring BFD

Bidirectional Forwarding Detection (BFD) provides a consistent failure detection method for network administrators, in addition to fast forwarding path failure detection. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable.

For more information about configuring BFD, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bfd/configuration/15-mt/irb-15-mt-book/irb-bi-fwd-det.html

Configuring Multi VRF

The Multi-VRF feature enables a service provider to support two or more Virtual Private Networks (VPNs), where the IP addresses can overlap several VPNs. The Multi-VRF Support feature uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each virtual routing and forwarding (VRF) instance. Interfaces in a VRF can be either physical, such as FastEthernet ports, or logical, such as VLAN switched virtual interfaces (SVIs), but a Layer 3 interface cannot belong to more than one VRF at any one time. The Multi-VRF Support feature allows an operator to support two or more routing domains on a customer edge (CE) device, with each routing domain having its own set of interfaces and its own set of routing and forwarding tables. The Multi-VRF Support feature makes it possible to extend the label switched paths (LSPs) to the CE and into each routing domain that the CE supports.

For more information about configuring Multi-VRF, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_pi/configuration/15-mt/iri-15-mt-book/mp-multi-vrf-vrf-lite.html

Configuring IPv6 Features

Internet Protocol version 6 (IPv6) expands the number of network address bits from 32 bits (in IPv4) to 128 bits, which can provide enough globally unique IP addresses for every networked device. The unlimited address space provided by IPv6 allows Cisco to deliver more and newer applications and services with reliability, improved user experience, and increased security.

For more information about configuring IPv6 addressing and basic connectivity, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_basic/configuration/15-mt/ip6b-15-mt-book.html