CHAPTER 4

Configuring the Cisco RAN Service Module with the Command-Line Interface

This chapter describes how to use the Cisco IOS software command-line interface (CLI) to configure the the Cisco RAN Service Module and includes the following sections:

For sample configurations, see Appendix B, "Configuration Examples".

For additional configuration topics, see the Cisco IOS configuration guide and command reference publications. These publications are available on the Documentation DVD that came with your router, available online at Cisco.com, or as printed copies that you can order separately.

Note If you skipped Chapter 2, "Cisco IOS Software Basics," and you have never configured a Cisco product, return to Chapter 2 and read it now. The chapter contains important information that you need to successfully configure your Cisco RAN Service Module.

Before You Begin

Before you configure the Cisco RAN Service Module, make sure the Cisco ONS 15454 platform is equipped with a proper software package and adequate hardware and interface cards. The software package release for the Cisco ONS 15454 should be at least 07.20-M06K-22.90 and up. And Cisco IOS release 12.2(29)SM, ransvc-ipran-mz image, must be installed on the Cisco RAN Service Module.

Verifying the Version of Cisco IOS Software

To implement the Cisco RAN Service Module in a RAN-O solution, Cisco IOS Release 12.2(29)SM must be installed on the Cisco ONS 15454. To verify the version of Cisco IOS software, use the **show version** command.

The **show version** command displays the configuration of the system hardware, the software version, the names and sources of the configuration files, and the boot images.

RAN Service Module Overview

The Cisco RAN Service Module is supported in the Cisco ONS 15454 chassis as one of the interface cards. There are four CPUs on the RAN Service Module card. One of these CPUs serves as a service CPU and the other three are traffic CPUs. The RAN Service Module interacts with the rest of the I/O interface cards in the Cisco ONS 15454 chassis through cross connect cards. The Cisco RAN Service Module has the following traffic interfaces:

- Four Gigabit Ethernet (GigE) interfaces: These interfaces are numbered GigE 0/0, GigE 1/0, GigE 2/0, and GigE 3/0. Each of these interfaces is assigned to one CPU. Interface GigE 0/0 is used for management traffic. The other GigE interfaces (GigE 1/0, GigE 2/0, and GigE 3/0) are used for backhaul communications. These interfaces do not interact with other I/O cards via the cross-connect, but rather are physical ports available on the faceplate of the RAN Service Module.
- Four POS interfaces: These interfaces are POS 0/0, POS 1/0, POS 2/0, and POS 3/0. Each interface resides on one CPU. Interface POS 0/0 is connected to the service CPU and it may be used for management traffic. The other POS interfaces are used for backhaul communications. These interfaces support HDLC and PPP encapsulation. In CTC, the POS interfaces are listed as STM-1 ports 5-8, and they can be cross-connected to other interface cards.
- Four ATM interfaces: Each CPU is equipped with its own ATM interface. Interface ATM0/0 is attached to the service CPU. And interfaces ATM1/0, ATM2/0, and ATM3/0 are connected to traffic CPUs 1, 2, and 3 respectively. These ATM interfaces are not directly accessible via CTC. They must first be assigned to one of four VC4 ports using an IOS-based cross-connect feature which is configured on the RAN Service Module itself. The 4 VC4 ports are listed as STM-1 ports 1-4 in the CTC card view. The IOS based cross-connect feature is described more fully in the section "Configuring the IOS Based Cross-connect.

- 126 E1/T1 interfaces: There are 42 of these interfaces assigned to each of the three traffic CPUs. The interface correspond to cross connect ports 9 through 134. The E1/T1interfaces serve as GSM-abis and backhaul (HDLC/PPP) connections. Users can configure up to 80 GSM-abis interfaces and 40 HDLC/PPP interfaces. No fractional E1/T1 is supported on the Cisco RAN Service Module, All time slots must be configured in a channel group.

Configuration Sequence

The following Summary of Steps section provides the recommended primary configuration sequence for the Cisco RAN Service Module. These steps have configuration sub-steps or tasks within the primary steps or tasks.

Note The installation of the Cisco RAN Service Module should be completed before attempting the configuration (see the "Related Documentation" section on page ix for more information).

The configuration sequence of the Cisco RAN Service Module assumes that you will have already had some familiarity with the configuration of Cisco products. It is also assumed that you are familiar with your own network configurations and that you are familiar with the Command Line Interface (CLI) used in configuring Cisco products.

Note For correct CLI syntax and format, see the "Cisco RAN Service Module Command Reference" section on page A-1.

Summary of Steps

Perform the following tasks to configure the Cisco RAN Service Module.

1. Configuring the Hostname and Password, page 4-4
2. Verifying the Hostname and Password, page 4-5
3. Configuring Gigabit Ethernet Interfaces, page 4-5
4. Configuring the POS Interfaces, page 4-7
5. Configuring the Backhaul Links, page 4-9
6. Configuring GSM-Abis Links, page 4-15
7. Configuring the IOS-based Cross-connect, page 4-17
8. Configuring UMTS Links, page 4-19
9. Configuring QoS, page 4-21
10. Configuring Redundancy, page 4-29
11. Configuring for SNMP Support, page 4-30
12. Configuring Graceful Degradation, page 4-33
13. Saving Configuration Changes, page 4-34

Configuring the Hostname and Password

Two important configuration tasks that you might want to perform first are to configure the hostname and to set an encrypted password. Configuring a host name allows you to distinguish multiple Cisco routers from each other. Setting an encrypted password allows you to prevent unauthorized configuration changes.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

To configure a hostname and to set an encrypted password, follow these steps:

Step 1 Enter enable mode.

```
Router> enable
```

The Password prompt appears. Enter your password.

```
Password: password
```

You have entered the enable mode when the prompt changes to `Router#`.

Step 2 Enter global configuration mode.

```
Router# configure terminal

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

You have entered global configuration mode when the prompt changes to `Router(config)#`.

```
Router(config)#
```

Step 3 Change the name of the router to a meaningful name. Substitute your hostname for `Router`.

```
Router(config)# hostname Router

Router(config)#
```

Enter an enable secret password. This password provides access to the privileged EXEC mode. When you type **enable** at the EXEC prompt (`Router>`), you must enter the enable secret password to access the configuration mode. Enter your secret password.

```
Router(config)# enable secret secret password
```

Step 4 Exit back to global configuration mode.

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

Verifying the Hostname and Password

To verify that you have correctly configured the hostname and password, follow these steps:

Step 1 Enter the **show config** command:

```
Router# show config
Using 1888 out of 126968 bytes
!
version XX.X
.
.
.
!
hostname Router
!
enable secret 5 $1$60L4$X2JYOwoDc0.kqa1loO/w8/
.
.
.
```

Check the hostname and encrypted password, which are displayed near the top of the command output.

Step 2 Exit global configuration mode and attempt to reenter it, using the new enable password:

```
Router# exit
.
.
.
Router con0 is now available
Press RETURN to get started.
Router> enable
Password: password
Router#
```

Configuring Gigabit Ethernet Interfaces

The Gigabit Ethernet interfaces are numbered GigE 0/0, GigE 1/0, GigE 2/0, and GigE 3/0. Each of these interfaces is assigned to one CPU. Interface GigE 0/0 is used for management traffic. The other GigE interfaces (GigE 1/0, GigE 2/0, and GigE 3/0) are used for backhaul communications. These interfaces do not interact with other I/O cards via the cross-connect, but rather are physical RJ-45 ports available on the faceplate of the RAN Service Module.

To configure the Gigabit Ethernet (GigE) interface on the Cisco RAN Service Module, complete the following tasks:

- Configuring the Gigabit Ethernet Interface IP Address
- Setting the Speed and Duplex Mode, page 4-6
- Enabling the Gigabit Ethernet Interface, page 4-7

Configuring the Gigabit Ethernet Interface IP Address

Use the following instructions to perform a basic IP Address configuration: specifying the port adapter, assigning an IP address and subnet mask for the interface.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

To configure the GigE interface, follow these steps, while in the global configuration mode:

Step 1 Specify the port adapter type and the location of the interface to be configured.

```
Router(config)# interface gigabitethernet cpu<0-3>/port<0-0>
```

Step 2 Assign an IP address and subnet mask to the interface.

```
Router(config-if)# ip address ip_address subnet_mask
```

Setting the Speed and Duplex Mode

The Gigabit Ethernet (GigE) ports of the Cisco RAN Service Module can run in full- or half- duplex mode and at 1000 Mbps, 100 Mbps, or 10 Mbps. The Cisco RAN Service Module has an auto-negotiation feature that allows the router to negotiate the speed and duplex mode with the corresponding interface at the other end of the connection.

Auto-negotiation is the default setting for the speed and transmission mode.

When configuring an interface speed and duplex mode, follow these guidelines:

- If both ends of the line support auto-negotiation, we highly recommend the default auto negotiation settings.
- When auto-negotiation is turned on for either speed or duplex mode, it auto- negotiates both speed and the duplex mode.
- If one interface supports auto-negotiation, and the interface at the other end does not, configure the duplex mode and speed on both interfaces. If you use the auto-negotiation setting on the supported side, the duplex mode setting will be set at half-duplex.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

To configure speed and duplex operation, follow these steps, while in the interface configuration mode:

Step 1 Specify the duplex operation.

```
Router(config-if)# duplex [auto | half | full]
```

Step 2 Specify the speed.

```
Router(config-if)# speed [auto | 1000 | 100 | 10]
```

Enabling the Gigabit Ethernet Interface

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

Once you have configured the Gigabit Ethernet (GigE) interface, enable it, by following this step, while in the interface configuration mode:

Step 1 Enable the interface.

```
Router(config-if)# no shutdown
```

Configuring the POS Interfaces

The POS interfaces are POS 0/0, POS 1/0, POS 2/0, and POS 3/0. Each interface resides on one CPU. Interface POS 0/0 is connected to the service CPU and it may be used for management traffic. The other POS interfaces are used for backhaul communications. These interfaces support HDLC and PPP encapsulation. In CTC, the POS interfaces are listed as STM-1 ports 5-8, and they can be cross-connected to other interface cards.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

To configure the POS interface on the Cisco RAN Service Module, complete the following tasks, while in the global configuration mode:

Step 1 Specify the port adapter type and the location of the interface to be configured. End the following command with a Ctrl/Z.

```
Router(config)# interface pos cpu<0-3>/port<0-0>
```

Step 2 Assign an IP address and subnet mask to the interface. End the following command with a Ctrl/Z.

```
Router(config-if)# ip address ip_address subnet_mask
```

Step 3 Assign the encapsulation type. End the following command with a Ctrl/Z.

```
Router(config-if)# encapsulation <encap type>
```

Step 4 Set the flag C2 byte. The default value of the C2 byte is 0x16. The C2 byte is the path signal label. The purpose of this byte is to communicate the payload type that the SONET Framing OverHead (FOH) encapsulates. The C2 byte allows a single interface to transport multiple payload types simultaneously. The C2 byte needs to be 0x16 for hdlc/ppp. End the following command with a Ctrl/Z.

```
Router(config-if)# pos flag c2 <byte value>
```

Step 5 Set the triggers for alarm generations. By default, all the following failure types trigger an alarm generation:

all - Path Signal Label Encapsulation Mismatch failure
encap - Path Signal Label Encapsulation Mismatch failure
pais - Path Alarm Indication Signal failure <default>
plmp - Path Label Mismatch failure <default>
plop - Path Loss of Pointer failure <default>
ppdi - Path Payload Defect Indication failure <default in lex encap>
prdi - Path Remote Defect Indication failure
puneq - Path Label Equivalent to Zero failure

The user can turn off any trigger by entering the **no** command in front of the trigger.

```
Router(config-if)# no pos trigger failure-types
```

Step 6 Set the trigger delay time in milli-seconds. The milli-second range is 200-2000. End the following command with a Ctrl/Z.

```
Router(config-if)# pos trigger delay milliseconds
```

Step 7 By default, POS scrambling is enabled on a Cisco RAN Service Module, and when enabled, scrambling is enabled on all POS interfaces. The command show interface pos <cpu>/<port> can be used to determine if scrambling is enabled on the RAN Service Module. The command pos-scrambling can be used to enable/disable scrambling on all POS interfaces.

Examples for this scrambling command follow:

Example 1: This command enables scrambling on all POS interfaces:

```
Router(config)# pos-scrambling
Router(config)# end
```

Example 2: This command disables scrambling on all POS interfaces:

```
Router(config)# no pos-scrambling
Router(config)# end
```

Example 3: This command shows if the state of POS scrambling:

```
Router# show interface pos cpu<0-3>/port<0-0>
Router# show interface pos 1/0
POS1/0 is down, line protocol is down
  Hardware is Packet over Sonet
  Internet address is 100.1.1.12/24
  MTU 1500 bytes, BW 155520 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 32, loopback not set
  Keepalive set (10 sec)
  Scramble enabled
  Last input 1d00h, output 1d00h, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
```

```
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
   45080 packets input, 2977622 bytes
   Received 38613 broadcasts (0 IP multicast)
   0 runts, 0 giants, 0 throttles
            0 parity
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   45187 packets output, 3078744 bytes, 0 underruns
   0 output errors, 0 applique, 23 interface resets
   0 output buffer failures, 0 output buffers swapped out
   41 carrier transitions
```

Configuring the Backhaul Links

To configure the backhaul links, complete the following tasks:

- Configuring Links to E1/T1 Traffic, this page
- Configuring E1 Controllers, page 4-10
- Configuring T1 Controllers, page 4-11
- Configuring Multilink Backhaul Interface, page 4-12
- Configuring the PPP Backhaul Interfaces, page 4-14

Configuring Links to E1/T1 Traffic

There are a total of 126 E1/T1 interfaces. There are 42 of these interfaces assigned to each of the three traffic CPUs. The interface correspond to cross connect ports 9 through 134. The E1/T1interfaces serve as GSM-abis and backhaul (HDLC/PPP) connections. Users can configure up to 80 GSM-abis interfaces and 40 HDLC/PPP interfaces. No fractional E1/T1 is supported on the Cisco RAN Service Module, All time slots must be configured in a channel group.

Use the following instructions to perform a basic interface configuration: enabling the module and enabling an interface.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

Step 1 Enter the enable mode.

```
Router> enable
```

Step 2 Enter the password.

```
Password: password
```

You have entered the enable mode when the prompt changes to `Router#`.

Step 3 Enter the global configuration mode.

```
Router# configure terminal
```

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

Router(config)#
```

You have entered the global configuration mode when the prompt changes to `Router(config)#`.

Note To see a list of the configuration commands available to you, enter **?** at the prompt or press the **Help** key while in the configuration mode.

Configuring E1 Controllers

Use the following instructions to perform a basic E1 controller configuration: specifying the E1 controller, specifying the channel-group, configuring the serial interface, configuring PPP encapsulation, and enabling keepalive packets. You might also need to enter other configuration commands, depending on the requirements for your system configuration and the protocols you plan to route on the interface.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

To configure the E1 controllers, follow these steps, while in the global configuration mode:

Step 1 Specify the controller that you want to configure. Controller E1 1/0 of the Cisco RAN Service Module maps to cross connect port 9. Controller E1 1/1 maps to port 10. Or the command **show controller E1** can be used to look up the cross connect port for the controller.

```
Router(config)# controller e1 cpu/port
```

For example, the following command configures the E1 controller on CPU 1, port 0:

```
Router(config)# controller e1 1/0
```

You have entered the controller configuration mode when the prompt changes to `Router(config-controller)#`.

Step 2 Specify the channel-group and time slots to be mapped. Once you configure a channel-group, the serial interface is automatically created.

```
Router(config-controller)# channel-group channel-no timeslots timeslot-list
```

- *channel-no*—ID number to identify the channel group. The valid range is 0 to 30.
- *timeslot-list*—Timeslots (DS0s) to include in this channel group. The valid timeslots are 1 to 31.

For example, the following command configures the channel-group and time slots for an E1 controller:

```
Router(config-controller)# channel-group 0 timeslots 1-31
```

Note When you are using the **channel-group** *channel-no* **timeslots** *timeslot-list* command to change the configuration of an installed card, you must enter the **no channel-group** *channel-no* **timeslots** *timeslot-list* command first. Then enter the **channel-group***channel-no* **timeslots** *timeslot-list*

Step 3 Exit the controller configuration mode.

```
Router(config-controller)# exit
```

Step 4 Configure the serial interface. Specify the CPU number, port, and channel-group number.

```
Router(config)# interface serial cpu/port:channel
Router(config-if)#
```

Note To see a list of the configuration commands available to you, enter **?** at the prompt or press the **Help** key while in the configuration mode.

Step 5 To configure PPP encapsulation, enter the following command:

```
Router(config-if)# encapsulation ppp
```

Step 6 Enable keepalive packets on the interface and specify the number of times keepalive packets will be sent without a response before bringing down the interface:

```
Router(config-if)# keepalive [period]
```

Step 7 Return to Step 1 to configure additional E1/T1 controllers.

Step 8 Exit the interface configuration mode.

```
Router(config-if)# exit
```

Configuring T1 Controllers

Use the following instructions to perform a basic T1 controller configuration: specifying the T1 controller, specifying the framing type, specifying the line code form, specifying the channel-group and time slots to be mapped, configuring the cable length, configuring the serial interface, configuring PPP encapsulation, and enabling keepalive packets. You might also need to enter other configuration commands, depending on the requirements for your system configuration and the protocols you plan to route on the interface.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

To configure the T1 interfaces, follow these steps, while in the global configuration mode:

Step 1 Specify the controller that you want to configure.

```
Router(config)# controller t1 cpu/port
```

Step 2 Specify the framing type.

```
Router(config-controller)# framing esf
```

Step 3 Exit controller configuration mode.

```
Router(config-controller)# exit
```

Step 4 Configure the serial interface. Specify the T1 CPU number, port number, and channel-group.

```
Router(config)# interface serial cpu/port:channel
```

Step 5 Enter the following command to configure PPP encapsulation.

```
Router(config-if)# encapsulation ppp
```

Step 6 Enable keepalive packets on the interface and specify the number of times that keepalive packets will be sent without a response before the interface is brought down:

```
Router(config-if)# keepalive [period]
```

Step 7 Return to Step 1 to configure additional T1 controllers.

Step 8 Exit to the global configuration mode.

```
Router(config-if)# exit
```

Configuring Multilink Backhaul Interface

A multilink interface is a special virtual interface that represents a multilink PPP bundle. The multilink interface coordinates the configuration of the bundled link, and presents a single object for the aggregate links. However, the individual PPP links that are aggregated must also be configured. Therefore, to enable multilink PPP on multiple serial interfaces, you first need to set up the multilink interface, and then configure each of the serial interfaces and add them to the same multilink interface.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

The Cisco RAN Service Module can support up to 10 E1 or T1 interfaces through the multilink interface.

Complete the following configuration tasks for a multilink backhaul interface.

- Creating a Multilink Bundle, this page
- Enable Real-Time Transport Protocol (RTP) Header-Compression, page 4-13

Creating a Multilink Bundle

To create a multilink bundle, follow these steps, while in the global configuration mode:

Step 1 Create a multilink bundle and enter the interface configuration mode:

```
Router(config)# interface multilink group-number
```

- *group-number*—Number of the multilink bundle.

For example, the following command creates a multilink bundle 5:

```
Router(config)# interface multilink5
Router(config-if)#
```

To remove a multilink bundle, use the **no** form of this command.

Note To see a list of the configuration commands available to you, enter **?** at the prompt or press the **Help** key while in the configuration mode.

Step 2 Assign an IP address to the multilink interface.

```
Router(config-if)# ip address address [subnet mask]
```

- *address*—The IP address.
- *subnet mask*—Network mask of IP address.

For example, the following command creates an IP address and subnet mask:

```
Router(config-if)# ip address 10.10.10.2 255.255.255.0
```

Step 3 Enable keepalive packets on the interface and specify the number of times the keepalive packets will be sent without a response before bringing down the interface.

```
Router(config-if)# keepalive [period]
```

- *period*—(Optional) Integer value in seconds greater than 0. The default is 10.

For example, the following command restricts (identifies) the multilink interface, 5, that can be negotiated:

```
Router(config-if)# keepalive 1
```

Enable Real-Time Transport Protocol (RTP) Header-Compression

To enable RTP Header Compression, follow these steps, while in the interface configuration mode:

Step 1 Enable RTP header-compression.

```
Router(config-if)# ip rtp header-compression [passive | iphc-format | ietf-format]
[periodic-refresh]
```

- **passive**—(Optional) Compresses outgoing RTP packets only if incoming RTP packets on the same interface are compressed. If you do not specify the passive keyword, all RTP packets are compressed. This option is not applicable on PPP links.
- **iphc-format**—(Optional) Indicates that the IP Header Compression (IPHC) format of header compression will be used.
- **ietf-format**—(Optional) Indicates that the Internet Engineering Task Force (IETF) format of header compression will be used.
- **periodic-refresh**—(Optional) Indicates that the compressed IP header will be refreshed periodically.

For example, the following command enables RTP header-compression in the Internet Engineering Task Force (IETF) format by suppressing the IP ID in the RTP/UDP header compression:

```
Router(config-if)# ip rtp header-compression ietf-format [periodic-refresh]
```

Configuring the PPP Backhaul Interfaces

Use the following instructions to perform a basic backhaul interface configuration: enabling an interface, configuring PPP encapsulation, enabling multilink PPP operation, and specifying an ID number for the multilink interface. You might also need to enter other configuration commands, depending on the requirements for your system configuration and the protocols you plan to route on the interface.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

To continue the configuration of the backhaul links for the E1 controllers, follow these steps, while in the global configuration mode:

Step 1 Configure the serial interface. Specify the CPU number, port number, and channel-group.

```
Router(config)# interface serial cpu/port: channel-group
```

Where:

- *cpu*—CPU number.
- *port*—Port number of the interface.
- *channel-group*—ID number to identify the channel group.

For example, the following command identifies the serial interface located in Cpu 1, port 0, channel-group 0:

```
Router(config)# interface serial1/0:0
Router(config-if)#
```

Note To see a list of the configuration commands available to you, enter **?** at the prompt or press the **Help** key while in the configuration mode.

Step 2 Do not assign an IP address and subnet mask to the interface.

```
Router(config-if)# no ip address ip_address subnet_mask
```

Step 3 To configure PPP encapsulation, enter the following command:

```
Router(config-if)# encapsulation ppp
```

Step 4 Enable multilink PPP operation.

```
Router(config-if)# ppp multilink
```

Step 5 Enable the interleaving of packets among the fragments of larger packets on the multilink ppp bundle.

```
Router(config-if)# ppp multilink interleave
```

Step 6 Specify the maximum configurable bandwidth, The default percent value is 75 percent.

```
Router(config-if)# max-reserved-bandwidth percent
```

Step 7 Specify an identification number for the multilink interface.

```
Router(config-if)# multilink-group group-number
```

- *group-number*—Multilink group number.

For example, the following command restricts (identifies) the multilink interface, 5, that can be negotiated:

```
Router(config-if)# multilink-group 5
```

Step 8 Enable keepalive packets on the interface and specify the number of times the keepalive packets will be sent without a response before bringing down the interface.

```
Router(config-if)# keepalive [period]
```

- *period*—(Optional) Integer value in seconds greater than 0. The default is 10.

For example, the following command indicates the number of times the keepalive packets will be sent as 1:

```
Router(config-if)# keepalive 1
```

Configuring GSM-Abis Links

Use the following instructions to perform a basic GSM-Abis configuration on the Cisco RAN Service Module, by entering the following Cisco IOS commands at the router prompt (see the "Understanding the Cisco RAN Service Module Interfaces" section on page 3-1 for information about slot and port numbering on the Cisco RAN Service Module). You might also need to enter other configuration commands, depending on the requirements for your system configuration and the protocols you plan to route on the interface.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

To configure the GSM-Abis attributes, follow these steps while in the global configuration mode:

Step 1 Specify the controller that you want to configure by entering the controller configuration mode.

```
Router(config)# controller e1 cpu/port
```

- *cpu*—CPU number.
- *port*—Number of the serial port.

```
Router(config)# controller e1 1/2
Router(config-controller)#
```

Step 2 Specify the channel-group and time slots to be mapped. Once you configure a channel-group, the serial interface is automatically created.

```
Router(config-controller)# channel-group channel-no timeslots timeslot-list speed {64}
```

- *channel-no*—ID number to identify the channel group. The valid range is 0 to 30.
- *timeslot-list*—Timeslots (DS0s) to include in this channel group. The valid timeslots are 1 to 31.
- **speed {64}**—The speed of the DS0: 64 kbps.

For example, the following command configures the channel-group and time slots for the E1 controller:

```
Router(config-controller)# channel-group 0 timeslots 1-31 speed 64
```

Note When you are using the **channel-group** *channel-no* **timeslots** *timeslot-list* **{64}** command to change the configuration of an installed card, you must enter the **no channel-group** *channel-no* **timeslots** *timeslot-list* **speed {64}** command first. Then enter the **channel-group** *channel-no* **timeslots** *timeslot-list* **{64}** command for the new configuration information.

Step 3 Exit back to global configuration mode.

```
Router(config-controller)# exit
```

Step 4 To Configure the GSM-Abis interface, first specify the serial interface that you want to configure by entering the interface configuration mode.

```
Router(config)# interface serial cpu/port:channel-group
```

- *cpu*—CPU number.
- *port*—Number of the port being configured.
- *channel-group*—Specifies the E1 channel group number defined with the channel-group controller configuration command.

For example, the following command enables the serial interface on CPU 1, port 2, channel group 0:

```
Router(config)# interface serial 1/2:0
Router(config-if)#
```

Note To see a list of the configuration commands available to you, enter **?** at the prompt or press the **Help** key while in the configuration mode.

Step 5 Enter the following command to configure GSM-Abis interface encapsulation in the interface configuration mode.

```
Router(config-if)# encapsulation gsm-abis
```

- **gsm-abis**—Type of interface layer.

For example, the following command enables encapsulation on the GSM-ABIS interface layer:

```
Router(config-if)# encapsulation gsm-abis
```

Step 6 To configure the local parameters required to establish an IP/UDP backhaul connection, enter the following command including the IP address and port you want to establish the IP/UDP backhaul connection from in the interface configuration mode.

```
Router(config-if)# gsm-abis local ip-address port
```

- *ip-address*—The IP address for the entry you wish to establish.
- port—The port you want to use for the entry you wish to establish.

For example, the following command configures the gsm-abis local parameters to an IP address of 10.10.10.2 located on port 5502:

```
Router(config-if)# gsm-abis local 10.10.10.2 5502
```

Step 7 To configure the remote parameters required to establish an IP/UDP backhaul connection, enter the following command including the IP address and port you want to establish the IP/UDP backhaul connection to in the interface configuration mode.

```
Router(config-if)# gsm-abis remote ip-address port
```

- *ip-address*—The IP address for the entry you wish to establish.
- *port*—The port you want to use for the entry you wish to establish.

For example, the following command configures the **gsm-abis remote** parameters to an IP address of 10.10.10.1 located on port 5502:

```
Router(config-if)# gsm-abis remote 10.10.10.1 5502
```

Step 8 Return to Step 1 to configure the additional gsm-abis links.

Step 9 Exit the interface configuration mode.

```
Router(config-if)# exit
```

Configuring the IOS-based Cross-connect

The RAN Service Module is equipped with an IOS-based cross connect feature which allows multiple ATM interfaces to be assigned to a single VC4 port. This enables the provisioner to connect all three traffic CPUs to a single STM-1 interface on the RNC. There is no default configuration for this feature, so it must be configured before UMTS links can be used.

Summery of Steps:

- Assign ATM interfaces to the desired VC4 port.
- Configure the number of VPI/VCI bits assigned to each VC4 Port.
- Activate the cross-connect configuration
- Configure VC4 port cell-payload scrambling settings (optional).
- Configure sts-stream scrambling settings (optional).

Step 1 Assign ATM interfaces to the desired VC4 port.

All four ATM interfaces can be assigned to a single VC4 port, or a single interface can be assigned to each VC4 port, or some combination thereof. No ATM interface can be added to more than one VC4 port:

```
Router(config)#cross-connect vc4 port VC4 port number
```

- *VC4 port number* - The number of the VC4 port. This corresponds to STM-1 ports 1-4 shown in the card view on CTC

```
Router(config-cc)#connect interface atm number/0
```

- *Slot* - The interface number of the ATM interface. A zero corresponds to the service CPU, And numbers 1-3 correspond to traffic CPUs 1-3.

For example, to assign all ATM interfaces to VC4 port 1:

```
Router(config)#cross-connect vc4 port 1
Router(config-cc)#connect interface atm 0/0
Router(config-cc)#connect interface atm 1/0
Router(config-cc)#connect interface atm 2/0
Router(config-cc)#connect interface atm 3/0
```

Step 2 Configure the number of VPI/VCI bits assigned to each VC4 Port.

The RAN Service Module supports the configuring of PVCs out of a pool of up to 2048 PVCs. The range of values permitted for the virtual path identifier (VPI) and virtual channel identifier (VCI) portions of the PVC identifier are determined by the command:

```
Router(config-cc)#max vpi-bits number vpi bits vci-bits number vci bits
```

- *number vpi bits* - The number of bits assigned for the VPI number. Supported ranges are 0-8. A zero indicates that the VPI number is always zero.
- *number vci bits* - The number of bits assigned for the VPI number. Supported ranges are 0-11.

For example, with the following configuration of VC port 4 would permit the VPI to be configured in the range 0-7 and the VCI to be configured in the range 0-255.

```
Router(config)#cross-connect vc4 port 1
Router(config-cc)#max vpi-bits 3 vci-bits 8
```

Note The fact that the VPI/VCI bits are configured along bit boundaries introduces some limitations in the provisioning of PVCs. For example, consider that you want to assign the interface ATM0/0 to VC4 port 1 for management traffic and interfaces ATM1/0, ATM2/0, and ATM3/0, to port 2. Even if only a few PVCs are required for VC4 port 1, the pool of PVCs assignable to VC4 port 2 would be reduced to 1024. Also, note that 2048 represents the only pool from which PVCs can be selected to be configured. The actual maximum number of PVCs which can actually be simultaneously configured is 255 PVCs per UMTS peer with a maximum of 649 per traffic CPU.

Note PVCs 0/3 and 0/4 are reserved PVCs and they cannot be configured.

Step 3 Activate the cross-connect configuration.

The following configuration command causes the above configurations to be activated on the RAN Service Module. Once this command is configured, any changes made to the ATM interface assignment to VC4 ports, or any changes to the max VPI or VCI bits will require a reload of the card to take effect. Once this card is configured and stored in the startup configuration, all IOS-based cross connect commands take effect at startup time.

```
Router(config)# ran-opt atm initialize
```

Step 4 Configure sts-stream scrambling settings (optional).

By default, the RAN Service Module uses STM-1 stream scrambling. To change this, use the global configuration command ran-opt atm scrambling. This changes the stream scrambling setting for VC4 ports 1-4. For example, to disable stream scrambling use the following command:

```
Router(config)# no ran-opt atm scrambling
```

Configuring UMTS Links

Use the following instructions to perform a basic UMTS-Iub configurational on the Cisco RAN Service Module, Enter the following Cisco IOS commands at the router prompt (see the "Understanding the Cisco RAN Service Module Interfaces" section on page 3-1 for information about slot and port numbering on the Cisco RAN Service Module). You might also need to enter other configuration commands, depending on the requirements for your system configuration and the protocols you plan to route on the interface.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

To configure the UMTS-Iub attributes, follow these steps while in the global configuration mode:

Step 1 Enter interface configuration mode and specify the location of the interface.

```
Router(config)# interface ATM cpu/port
```

- *cpu*—Specifies the CPU.
- *port*—Specifies the port.

For example, the following command specifies the location of the interface as ATM 1/0.

```
Router(config# interface atm1/0
```

Note To see a list of the configuration commands available to you, enter **?** at the prompt or press the **Help** key while in the configuration mode.

Step 2 Use the aggnode command to configure the interface as an aggregate node mode.

```
Router(config-if)# atm umts-iub [aggnode]
```

For example: Since the RAN-SVC module will be used as an aggregation node, use the following configuration:

```
Router(config-if)# atm umts-iub aggnode
```

Step 3 In aggregation node mode, UMTS peers are configured on subinterfaces. To select a subinterface, use the command,

```
Router(config-if)# interface ATM cpu/port.subinterface
```

For example: To configure a UMTS peer on interface ATM1/0.10:

```
Router(config-if)# interface ATM1/0.10
```

Step 4 To configure the local parameters required to establish an IP/UDP backhaul connection, enter the following command including the IP address and port you want to establish the IP/UDP backhaul connection from.

```
Router(config-subif)# umts-iub local ip-address port
```

Step 5 To configure the remote parameters required to establish an IP/UDP backhaul connection, enter the following command including the IP address and port you want to establish the IP/UDP backhaul connection from.

```
Router(config-subif)# umts-iub remote ip-address port
```

Step 6 Create an ATM permanent virtual circuit (PVC):

```
Router(config-if)# pvc [name] vpi/vci vci [qsaal]
```

- *name*—(Optional) specifies the name of the ATM PVC interface you create.
- *vpi*—Specifies the ATM network virtual path identifier (VPI) of this PVC.
- *vci*—Specifies the ATM network virtual channel identifier (VCI) of this PVC.
- **qsaal**—(Optional) specifies the ATM adaptation layer as AAL5.

Note Typically AAL5 PVCs are defined using qsaal encapsulation. However, if the traffic profile is such that the AAL5 packets exceed normal signaling (272 bytes) payload size, it is recommended that the PVC be defined using AAL0.

This is commonly true for OAM PVCs and synchronization PVCs. NodeB Application Part (NBAP) and Access Link Control Application Part (ALCAP) PVCs can be defined using qsaal encapsulation.

For example, the following command specifies the ATM PVC interface with a VPI of 0 and a VCI of 100:

```
Router(config-if)# pvc 0/100
```

Note PVC definitions should match those on the NodeB and use the following definitions:

NBAP signaling – use qsaal
ALCAP signaling – use qsaal
AAL2 bearer – use encapsulation aal0
All other PVCs should use encapsulation aal0

Class of service should be defined to match the NodeB PVC class of service definitions. For instance, if the NodeB has defined a PVC with CBR, the PVC on the Cisco MWR 1941-DC-A router should use the same CBR definitions.

OAM can be defined on the PVCs as well. If the NodeB has OAM enabled on its PVC, OAM should be defined on the PVCs of the Cisco MWR 1941-DC-A router as well.

Step 7 Configure the ATM adaptation layer (AAL) and encapsulation type to AAL0 encapsulation.

```
Router(config-if-atm-vc)# encapsulation aal-encap
```

- *aal-encap*—Specifies the ATM adaptation layer (AAL) and encapsulation type

For example, the following command specifies the ATM adaptation layer (AAL) as AAL0:

```
Router(config-if)# encapsulation aal0
```

Step 8 Create another ATM permanent virtual circuit (PVC):

```
Router(config-subif)# pvc [name] vpi/vci vci [qsaal]
```

- *name*—(Optional) specifies the name of the ATM PVC interface you create.
- *vpi*—Specifies the ATM network virtual path identifier (VPI) of this PVC.
- *vci*—Specifies the ATM network virtual channel identifier (VCI) of this PVC.

- **qsaal**—(Optional) specifies the ATM adaptation layer as AAL5.

For example, the following command specifies the ATM PVC interface with a VPI of 0, a VCI of 100, and a QSAAL:

```
Router(config-if)# pvc 0/200 qsaal
```

Step 9 Return to Step 1 to configure additional interfaces.

Step 10 Exit the interface configuration mode.

```
Router(config-if)# exit
```

Configuring QoS

The RAN Services module supports the Low Latency Queuing (LLQ) feature. This feature feature brings strict priority queueing to Class-Based Weighted Fair Queueing (CBWFQ). Strict priority queueing allows delay-sensitive data such as voice to be dequeued and sent first (before packets in other queues are dequeued), giving delay-sensitive data preferential treatment over other traffic. The first step in configuring QOS on the RAN Services module is to classify traffic that is destined for the priority queue. The RAN Services module provides two methods for accomplishing this. First, it is possible to identify priority queue traffic by matching against the input interface. This method is cumbersome and requires adding additional match statements for each shorthaul interface. As new shorthaul interfaces are provisioned, match statements must be added to the class-map for the interfaces. The module supports a second method for identifying packets destined for the priority queue: matching against the differentiated services code point (DSCP), In this method the GSM and UMTS applications tag backhaul packets with a configured DSCP value. Because the same DSCP value can be configured for both GSM and UMTS, only a single match statement is required to classify traffic, and no changes need to be made to the class-map when new links are provisioned. The default value for both applications is express forwarding (ef).

Three new commands are added using the Interface Configuration mode for this new feature: **umts-iub set dscp**, **umts-iub set peering dscp**, and **gsm-abis set dscp** and one new ATM-VC InterfaceConfiguration command: **umts-iub set dscp** (see Appendix A, "Cisco RAN Service Module Command Reference" for detailed command information). These new commands allow you to perform the following:

- on the UMTS Shorthaul Interface
  - Set the default DSCP value with which to tag UMTS backhaul packets. Separate values can be assigned to backhaul packets containing data from the UMTS Shorthaul Interface and assigned to backhaul packets which contain peering information for the UMTS peers running on IOS.
- on the PVC of a UMTS Shorthaul Interface
  - DSCP values configured at the interface level will be applied by default to data from all PVCs. A separate DSCP may also be assigned to specific PVCs. This value supersedes the value configured at the interface level.
- on the GSM Shorthaul Interface
  - Set the DSCP value in such a way as to tag all the backhaul packets generated from the shorthaul in the GSM Abis interface.

In the following procedures, PVC 2/1 of ATM 1/0 will go to the priority queue and PVC 2/2 of ATM 1/0 will be considered the best effort traffic and will go to the Weighted Fair Queue.

Note Defining the **dscp** value under the PVC affects the way the ATM cells are bundled together as a backhaul. The more **dscp** values that are defined, the more limitations on how the ATM cells can be bundled. This, as a result, could affect backhaul efficiency. We recommend that you define at most two different **dscp** values for each shorthaul. One for llq traffic, and the other for best effort traffic.

Creating a Class Map

For each class map that you want to create, follow these steps, while in global configuration mode:

Step 1 Assign a name to your class map.

```
Router(config)# class-map [match-all | match-any] class_name
```

Where **match-any** means that a single match rule is sufficient for class membership and **match-all** means that only packets that have all the specified attributes are part of the class.

For example, the following command specifies the class map as an llq-class:

```
Router(config)# class-map match-any llq-class
```

When you enter the **class-map** command, you are in the class map configuration mode.

Step 2 To identify a specific IP differentiated service code point (DSCP) value as a match criterion, use the following command:

```
Router(config-cmap)# match ip dscp value
```

- **match ip dscp** *value* Specifies the exact value from 0 to 63 used to identify an IP DSCP value.

For example, the following command specifies cs2 to be used as a match criterion:

```
Router(config-cmap)# match ip dscp ef
```

For more information about this command, see the *Cisco IOS Quality of Service Solutions Command Reference* for your Cisco IOS Release.

Step 3 Exit the class map configuration mode.

```
Router(config-cmap)# exit
```

Creating a Policy Map

To create a policy map, follow these steps, while in the global configuration mode:

Step 1 Assign a name to your policy map.

```
Router(config)# policy-map policy_name
```

- *policy_name*— Specifies the name of the traffic policy. The traffic policy may contain one or more traffic classes.

For example, the following command specifies the policy map of low latency queueing (LLQ).

```
Router(config)# policy-map llq-policy
```

When you enter the **policy-map** command, you are in the policy map configuration mode.

Step 2 Associate the llq-policy with a class map.

```
Router(config-pmap)# class class_name
```

- *class_name*— Specifies the name of a traffic class you want to modify.

Specify the same *class_name* as you did in Step 1in the "Creating a Class Map" section on page 4-22.

For example, the following command specifies the class as the llq-class.

```
Router(config-pmap)# class llq-class
```

When you enter the **class** command, you are in the class submode of the policy-map configuration mode.

Step 3 Allocate a percentage of bandwidth to be used for the prority queue.

```
Router(config-pmap-c)# priority percent number
```

For example, the following command specifies a **priority percent** number of 99.

```
Router(config-pmap-c)# priority percent 99
```

Step 4 Associate the llq-policy with a default class map. The default class is used for non-priority traffic.

```
Router(config-pmap-c)# class class-default
```

Step 5 Allocate the remaining bandwidth to the default class.

```
Router(config-pmap-c)# bandwitdh remaining percent number
```

For example, the following command specifies the remaining bandwidth as 1 percent.

```
Router(config-pmap-c)# bandwidth remaining percent 1
```

Step 6 Limit the queue depth of the default queue.

```
Router(config-pmap-c)# queue-limit number
```

For example, the following command limits the queue depth to 45.

```
Router(config-pmap-c)# queue-limit 45
```

Note The queue limit on the default class should be less than the hold-queue specified on the multilink interface.

Step 7 Exit the class map and policy map configuration modes.

```
Router(config-pmap-c)# exit
Router(config-pmap)# exit
```

For more information about these commands, see the *Cisco IOS Quality of Service Solutions Command Reference* for your Cisco IOS Release.

Assigning GSM DSCP Values

Step 1 To assign the GSM DSCP values, first specify the serial interface that you want to configure by entering the interface configuration mode.

```
Router(config)# interface serial cpu/port:channel-group
```

For example, the following command enables the serial interface on CPU 1, port 2, channel group 0:

```
Router(config)# interface serial 1/2:0
Router(config-if)#
```

Step 2 To set the GSM DSCP value used as the interface default DSCP value to tag the backhaul packet, use the following command:

```
Router(config-if)# gsm set dscp value
```

- *value*—A number chosen to represent that packet of traffic.

For example, the following command specifies the number 16 for the packet of traffic for the umts-iub interface:

```
Router(config-if)# gsm set dscp 46
```

Assigning UMTS DSCP Values

Step 1 Enter the interface configuration mode and specify the location of the interface.

```
Router(config)# interface atm cpu/port
```

For example, the following command specifies the location of the interface as ATM 1/0.

```
Router(config# interface atm1/0
```

Step 2 Disable the IP address configuration for the physical layer interface.

```
Router(config-if)# no ip address
```

Step 3 Create an ATM path on the UMTS Iub interface, enter the following command:

```
Router(config-if)# atm umts-iub
```

Step 4 Disable the Interim Local Management Interface (ILMI) keepalive parameters.

```
Router(config-if)# interface atm 1/0.1 multipoint
```

Step 5 Create an ATM permanent virtual circuit (PVC):

```
Router(config-subif)# pvc [[name][vpi/vci][vci][qsaal]]
```

- *name*—(Optional) specifies the name of the ATM PVC interface you create.
- *vpi*—Specifies the ATM network virtual path identifier (VPI) of this PVC.
- *vci*—Specifies the ATM network virtual channel identifier (VCI) of this PVC.
- **qsaal**—(Optional) specifies the ATM adaptation layer as AAL5.

Note Typically AAL5 PVCs are defined using qsaal encapsulation. However, if the traffic profile is such that the AAL5 packets exceed normal signaling (272 bytes) payload size, it is recommended that the PVC be defined using AAL0.

This is commonly true for OAM PVCs and synchronization PVCs. NodeB Application Part (NBAP) and Access Link Control Application Part (ALCAP) PVCs can be defined using qsaal encapsulation.

For example, the following command specifies the ATM PVC interface with a VPI of 2and a VCI of 1:

```
Router(config-if)# pvc 2/1
```

Note PVC definitions should match those on the NodeB and use the following definitions:

NBAP signaling – use qsaal
ALCAP signaling – use qsaal
AAL2 bearer – use encapsulation aal0
All other PVCs should use encapsulation aal0

Class of service should be defined to match the NodeB PVC class of service definitions. For instance, if the NodeB has defined a PVC with CBR, the PVC on the Cisco RAN Service Module should use the same CBR definitions.

OAM can be defined on the PVCs as well. If the NodeB has OAM enabled on its PVC, OAM should be defined on the PVCs of the Cisco RAN Service Module as well.

Step 6 Configure the ATM adaptation layer (AAL) and encapsulation type to AAL0 encapsulation.

```
Router(config-if-atm-vc)# encapsulation aal-encap
```

- *aal-encap*—Specifies the ATM adaptation layer (AAL) and encapsulation type

For example, the following command specifies the ATM adaptation layer (AAL) as AAL0:

```
Router(config-if)# encapsulation aal0
```

Step 7 To set the DSCP value used as the interface default DSCP value to tag the backhaul packet, use the following command:

```
Router(config-if-atm-vc)# umts-iub set dscp value
```

- *value*—A number chosen to represent that packet of traffic.

For example, the following command specifies the number 16 for the packet of traffic for the umts-iub interface:

```
Router(config-if)# umts-iub set dscp 46
```

Step 8 Perform Steps 5 through 7 to set another PVC 2/2 with a umts-iub interface DSCP of 8.

Step 9 To overwrite the previous PVC 2/1 with a umts-iub interface DSCP of 16, use the following command:

```
Router(config-if)# umts-iub set dscp value
```

- *value*—A number chosen to represent that packet of traffic.

For example, the following command overwrites the number 16 for the packet of traffic for the umts-iub interface:

```
Router(config-if-atm-vc)# umts-iub set dscp 16
```

Step 10 Perform Steps 1 to 7 for ATM0/1 with a UMTS DSCP of 8.

Step 11 To overwrite the previous PVC 2/1 with a umts-iub interface DSCP of 16, use the following command:

```
Router(config-if-atm-vc)# umts-iub set dscp value
```

- *value*—A number chosen to represent that packet of traffic.

For example, the following command overwrites the number 16 for the packet of traffic for the umts-iub interface:

```
Router(config-if-atm-vc)# umts-iub set dscp 16
```

Step 12 Exit the interface configuration mode.

```
Router(config-subif)# exit
```

Assigning a QoS Boilerplate to an Interface

Use the following instructions to assign a QoS boilerplate to an interface: enabling a multilink interface, enable real-time packet interleaving, specifying an ID number for the multilink interface, configuring a maximum fragment size, enabling MCMP, specifying the percent of the interface bandwidth, and assigning the Qos boilerplate. You might also need to enter other configuration commands, depending on the requirements for your system configuration and the protocols you plan to route on the interface.

Step 1 Create a multilink bundle and enter the interface configuration mode:

```
Router(config)# interface multilink group-number
```

- *group-number*—Number of the multilink bundle.

For example, the following command creates a multilink bundle 5:

```
Router(config)# interface multilink5
Router(config-if)#
```

To remove a multilink bundle, use the **no** form of this command.

Step 2 Enable Transmission Control Protocol (TCP) header compression.

```
Router(config-if)# ip tcp header-compression keyword
```

For example, the following command enables IETF-Format as the header compression:

```
Router(config-if)# ip tcp header-compression ietf-format
```

Step 3 Disable the Cisco Discovery (CDP) on the interface.

```
Router(config-if)# no cdp enable
```

Step 4 By default, PFC handling is not enabled. Enter the following command to configure PFC on the router:

```
Router(config-if)# ppp pfc local {request | forbid}
```

Where:

- **request**—The PFC option is included in outbound configuration requests.
- **forbid**—The PFC option is not sent in outbound configuration requests, and requests from a remote peer to add the PFC option are not accepted.

For example, the following command creates how the router handles PFC:

```
Router(config-if)# ppp pfc local request
```

Step 5 To configure how the router handles the PFC option in configuration requests received from a remote peer, enter the following command:

```
Router(config-if)# ppp pfc remote {apply | reject | ignore}
```

Where:

- **apply**—PFC options are accepted and ACFC may be performed on frames sent to the remote peer.
- **reject**—PFC options are explicitly ignored.
- **ignore**—PFC options are accepted, but ACFC is not performed on frames sent to the remote peer.

For example, the following command allows PFC options to be accepted:

```
Router(config)# ppp pfc remote apply
```

Step 6 By default, ACFC handling is not enabled. To configure how the router handles ACFC in its outbound configuration requests, enter the following command:

```
Router(config-if)# ppp acfc local {request | forbid}
```

Where:

- **request**—The ACFC option is included in outbound configuration requests.
- **forbid**—The ACFC option is not sent in outbound configuration requests, and requests from a remote peer to add the ACFC option are not accepted.

For example, the following command creates how the router handles ACFC:

```
Router(config-if)# ppp acfc local request
```

Step 7 To configure how the router handles the ACFC option in configuration requests received from a remote peer, enter the following command:

```
Router(config-if)# ppp acfc remote {apply | reject | ignore}
```

Where:

- **apply**—ACFC options are accepted and ACFC may be performed on frames sent to the remote peer.
- **reject**—ACFC options are explicitly ignored.
- **ignore**—ACFC options are accepted, but ACFC is not performed on frames sent to the remote peer.

For example, the following command allows ACFC options to be accepted:

```
Router(config-if)# ppp acfc remote apply
```

Step 8 Enable multilink PPP operation.

```
Router(config-if)# ppp multilink
```

Step 9 Enable real-time packet interleaving.

```
Router(config-if)# ppp multilink interleave
```

Step 10 Specify an identification number for the multilink interface.

```
Router(config-if)# ppp multilink group group-number
```

- *group-number*—Multilink group number.

For example, the following command restricts (identifies) the multilink interface, 2, that can be negotiated:

```
Router(config-if)# ppp multilink group 2
```

Step 11 Enable multiclass multilink PPP (MCMP).

```
Router(config-if)# ppp multilink multiclass
```

Step 12 Specify the percent of the interface bandwidth allocated for LLQ.

```
Router(config-if)# max-reserved-bandwith percent
```

- *percent*—Percent of interface bandwidth allocated for LLQ.

For example, the following command specifies the interface bandwidth allocated for LLQ as 100%:

```
Router(config-if)# max-reserved-bandwidth 100
```

Step 13 Assign the QoS boilerplate to the multilink interface.

```
Router(config-if)# service-policy output policy_name
```

- *policy_name*— LLQ.

For example, the following command assigns the QoS boilerplate to the multilink interface policy name LLQ:

```
Router(config-if)# service-policy output llq-policy
```

Step 14 Set the size of the output queue.

```
Router(config-if)# hold-queue size in | out
```

- *size*— Number of packets held in the queue.
- *in* | *out*—Direction of packets being held, either input or output.

For example, the following command sets the size of the queue for the outbound packets at 50:

```
Router(config-if)# hold-queue 50 out
```

Note Specify a **hold-queue** limit. The limit needs to be greater than the **hold-queue** depth that is defined on the default class (see the "Creating a Class Map" section on page 4-22 for more information).

Step 15 Enable Transmission Control Protocol (TCP) header compression.

```
Router(config-if)# ip tcp header-compression keyword
```

For example, the following command enables IETF-Format as the header compression:

```
Router(config-if)# ip tcp header-compression ietf-format
```

Configuring Redundancy

With the exception of the Gigabit Ethernet interface, there is no IOS configuration to be configured on the Cisco RAN Service Module. The redundancy support for the RAN Service Module is 1:N, and it is configured from CTC. The configuration on the CTC can have either one protection group with one protect card and up to seven working cards, or it can have two protection groups with one protect card and up to four working cards in each group. The revertive timer can be disabled for the Cisco RAN Service Module so a user can manually switch back during a maintenance window.

The following is a brief explanation of redundancy support for the RAN Service Module. The protect card is running and has stored copies of the configurations for each working card in its protection group. In the event of a failure on a working card, the protect card activates the corresponding configuration that it has stored. The IOS configuration on the protect card should not be modified because the protect card needs to have a clean configuration to be ready to pick up the configuration from any of the working cards in the protection group when needed. After the working card recovers, services may be reverted to the working card. After reversion occurs, the protect card resets itself to clear out the configuration and to prepare to take over in case any other card in the protection group fails.

Redundancy on the Gigabit Ethernet interface is handled as part of the same mechanism described above. There is no separate mechanism such as HSRP that needs to be configured. In the event of a failure, the standby card configures the Gigabit Ethernet interface with the same IP as the working card. However, this presents a problem in that all layer-2 adjacent devices have the layer-2 address of the working card in their ARP tables. In order to make the transition from the working card to protect card seamless, a MAC address should be configured on the Gigabit Ethernet interfaces. When the protect card activates the configuration, it will configure the MAC address of the working card. One recommendation is to configure the MAC address that is physically assigned to the Gigabit Ethernet interface. This ensures that all MAC addresses remain unique. The following example shows how to configure the physical MAC address already assigned to the interface so that it will be stored in the configuration activated by the standby card.

Step 1 Determine the physically assigned MAC address of the Gigabit Ethernet interface which is in use:

```
Router#show interfaces GigabitEthernet 0/0 | i MAC
  Hardware is BCM1255 Internal MAC, address is 0006.0052.5300 (bia 0006.0052.5300)
```

Step 2 Configure the MAC Address on the interface

```
Router(config)#interface GigabitEthernet 0/0
Router(config-if)#mac-address 0006.0052.5300
```

Configuring for SNMP Support

Use the following instructions to configure for SNMP support: setting up the community access, establishing a message queue for each trap host, enabling the router to send SNMP traps, enabling SNMP traps for alarms, and enabling SNMP traps for a specific environment. You might also need to enter other configuration commands, depending on the requirements for your system configuration and the protocols you plan to route on the interface.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

To configure a Cisco RAN Service Module for SNMP, follow these steps while in the global configuration mode:

Step 1 To set up the community access string to permit access to the SNMP, use the **snmp-server community** command. The **no** form of this command removes the specified community string.

```
Router(config)# snmp-server community string [view view-name] [ro | rw] [number]
```

- *string*—Community string that acts like a password and permits access to the SNMP protocol.
- **view** *view-name*—(Optional) Name of a previously defined view. The view defines the objects available to the community.
- **ro**—(Optional) Specifies read-only access. Authorized management stations are only able to retrieve MIB objects.
- **rw**—(Optional) Specifies read-write access. Authorized management stations are able to both retrieve and modify MIB objects.
- *number*—(Optional) Integer from 1 to 99 that specifies an access list of IP addresses that are allowed to use the community string to gain access to the SNMP agent.

For example, the following command sets up the community access string as xxxxx with read-only access:

```
Router(config)# snmp-server community xxxxx RO
```

Step 2 To establish the message queue length for each trap host, use the **snmp-server queue-length** command.

```
Router(config)# snmp-server queue-length length
```

- *length*—Integer that specifies the number of trap events that can be held before the queue must be emptied.

For example, the following command establishes the number of trap events to 100:

```
Router(config)# snmp-server queue-length 100
```

Step 3 To enable the router to send SNMP traps or informs (SNMP notifications), use the **snmp-server enable traps** command. Use the **no** form of this command to disable SNMP notifications.

```
Router(config)# snmp-server enable traps [notification-type] [notification-option]
```

- ***notification-type*—snmp [authentication]**—Enables RFC 1157 SNMP notifications. Note that use of the **authentication** keyword produces the same effect as not using the **authentication** keyword. Both the **snmp-server enable traps snmp** and **snmp-server enable traps snmp authentication** forms of this command will globally enable (or, if using the **no** form, disable) the following SNMP traps:

  - authentication failure
  - linkup
  - linkdown
  - coldstart
  - warmstart

- ***notification-option***—(Optional) **atm pvc** [**interval** *seconds*] [**fail-interval** *seconds*]—The optional interval seconds keyword/argument combination specifies the minimum period between successive traps, in the range from 1 to 3600. Generation of PVC traps is dampened by the notification interval in order to prevent trap storms. No traps are sent until the interval lapses. The default interval is 30.

  The optional fail-interval seconds keyword/argument combination specifies the minimum period for storing the failed time stamp, in the range from 0 to 3600. The default fail-interval is 0.

  envmon [**voltage** | **shutdown** | **supply** | **fan** | **temperature**]—When the **envmon** keyword is used, you can enable a specific environmental notification type, or accept all notification types from the environmental monitor system. If no option is specified, all environmental notifications are enabled. The option can be one or more of the following keywords: **voltage**, **shutdown**, **supply**, **fan**, and **temperature**.

  isdn [**call-information** | **isdn u-interface**]—When the **isdn** keyword is used, you can specify the **call-information** keyword to enable an SNMP ISDN call information notification for the ISDN MIB subsystem, or you can specify the **isdnu-interface** keyword to enable an SNMP ISDN U interface notification for the ISDN U interface MIB subsystem.

  repeater [**health** | **reset**]—When the repeater keyword is used, you can specify the repeater option. If no option is specified, all repeater notifications are enabled. The option can be one or more of the following keywords:

  - **health**—Enables IETF Repeater Hub MIB (RFC 1516) health notification.
  - **reset**—Enables IETF Repeater Hub MIB (RFC 1516) reset notification.

For example, the following command enables traps for SNMP link down, link up, coldstart and warmstart:

```
Router(config)# snmp-server enable traps snmp linkdown linkup coldstart warmstart
```

Step 4 To enable SNMP traps for all IP-RAN notifications, enter:

```
Router(config)# snmp-server enable traps ipran
```

Note Besides enabling SNMP traps for all IP-RAN notifications, you can also enable traps for IP-RAN GSM alarms, UMTS alarms, and general information about the backhaul utilization (see Appendix A, "Cisco RAN Service Module Command Reference" for descriptions on how to use these SNMP commands.

Step 5 To enable SNMP traps for a specific environment, enter:

```
Router(config)# snmp-server enable traps envmon
```

Step 6 To specify the recipient of an SNMP notification operation, use the **snmp-server host** command. To remove the specified host, use the **no** form of this command.

```
Router(config)# snmp-server host host-addr [traps | informs] [version {1 | 2c | 3 [auth |
noauth | priv]}] community-string [udp-port port] [notification-type]
```

- *host-addr*—Name or Internet address of the host (the targeted recipient).

- **traps**—(Optional) Send SNMP traps to this host. This is the default.
- **informs**—(Optional) Send SNMP informs to this host.
- **version**—(Optional) Version of the Simple Network Management Protocol (SNMP) used to send the traps. Version 3 is the most secure model, as it allows packet encryption with the **priv** keyword. If you use the version keyword, one of the following must be specified:
  - **1**—SNMPv1. This option is not available with informs.
  - **2c**—SNMPv2C.
  - **3**—SNMPv3. The following three optional keywords can follow the version 3 keyword:

    –**auth** (Optional). Enables Message Digest 5 (MD5) and Secure Hash Algorithm (SHA) packet authentication

    –**noauth** (Default). The noAuthNoPriv security level. This is the default if the [auth | noauth | priv] keyword choice is not specified.

    –**priv** (Optional). Enables Data Encryption Standard (DES) packet encryption (also called "privacy").
- *community-string*—Password-like community string sent with the notification operation. Though you can set this string using the **snmp-server host** command by itself, we recommend you define this string using the **snmp-server community** command before using the **snmp-server host** command.
- **udp-port** *port*—UDP port of the host to use. The default is 162.
- *notification-type*—(Optional) Type of notification to be sent to the host. If no type is specified, all notifications are sent. The notification type can be one or more of the following keywords:
  - **bgp**—Sends Border Gateway Protocol (BGP) state change notifications.
  - **config**—Sends configuration notifications.
  - **dspu**—Sends downstream physical unit (DSPU) notifications.
  - **entity**—Sends Entity MIB modification notifications.
  - **envmon**—Sends Cisco enterprise-specific environmental monitor notifications when an environmental threshold is exceeded.
  - **frame-relay**—Sends Frame Relay notifications.
  - **hsrp**—Sends Hot Standby Routing Protocol (HSRP) notifications.
  - **isdn**—Sends Integrated Services Digital Network (ISDN) notifications.
  - **llc2**—Sends Logical Link Control, type 2 (LLC2) notifications.
  - **repeater**—Sends standard repeater (hub) notifications.
  - **rsrb**—Sends remote source-route bridging (RSRB) notifications.
  - **rsvp**—Sends Resource Reservation Protocol (RSVP) notifications.
  - **rtr**—Sends SA Agent (RTR) notifications.
  - **sdlc**—Sends Synchronous Data Link Control (SDLC) notifications.
  - **sdllc**—Sends SDLLC notifications.
  - **snmp**—Sends Simple Network Management Protocol (SNMP) notifications (as defined in RFC 1157).
  - **stun**—Sends serial tunnel (STUN) notifications.

- **syslog**—Sends error message notifications (Cisco Syslog MIB). Specify the level of messages to be sent with the **logging history level** command.
- **tty**—Sends Cisco enterprise-specific notifications when a Transmission Control Protocol (TCP) connection closes.
- **x25**—Sends X.25 event notifications.

For example, the following command specifies a recipient of the SNMP operation with a host-address of 10.20.30.40 with a version SNMP of SNMPv2C:

```
Router(config)# snmp-server host 10.20.30.40 version 2c
```

Step 7 Exit the global configuration mode.

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

Configuring Graceful Degradation

Congestion on the backhaul is detected by measuring its transmit jitter buffer level. If the transmit jitter buffer shrinks, it means that the backhaul packets are not arriving fast enough to fill the transmit jitter buffer indicating congestion. You should set the congestion abatement detection level at which a remote router will stop suppressing these timeslots.

Use the following instructions to configure graceful degradation by entering the following Cisco IOS commands at the router prompt.

You might also need to enter other configuration commands, depending on the requirements for your system configuration and the protocols you plan to route on the interface.

Note In the following procedure, press the **Return** key after each step unless otherwise noted. At any time, you can exit the privileged level and return to the user level by entering **disable** at the `Router#` prompt.

To configure graceful degradation, follow these steps while in the global configuration mode:

Step 1 Perform Steps 1 through 10 as described in the previous procedure (see the "Configuring GSM-Abis Links" procedure on page 4-15).

Step 2 To set the congestion detection algorithm to monitor the transmit jitter buffer so as to send the congestion indicator signals to the remote when the congestion is detected, enter the following command.

```
Router(config-if)# gsm-abis congestion enable
```

Step 3 To set the congestion abate detection level, enter the following command.

```
Router(config-if)# gsm-abis congestion abate ms
```

- *ms*—The value of the congestion abate in milliseconds.

For example, the following command configures the **gsm-abis congestion abate** detection level to a value 250 ms:

```
Router(config-if)# gsm-abis congestion abate 250
```

Note The abate detection level is defined as x milliseconds of continuous congestion abatement (that is, no congestion indications).

Step 4 To set the congestion onset detection level at which the remote router will start suppressing all timeslots that are not defined as critical in an effort to alleviate the congestion, enter the following command.

```
Router(config-if)# gsm-abis congestion onset ms
```

- *ms*—The value of the congestion onset in milliseconds.

For example, the following command configures the **gsm-abis congestion onset** detection level to a value 100 ms:

```
Router(config-if)# gsm-abis congestion onset 100
```

Note The onset detection level is defined as x milliseconds of continuous congestion detected.

Step 5 To define the critical timeslots that are exempt from suppression during congestion onset, enter the following command.

```
Router(config-if)# gsm-abis congestion critical timeslot-range
```

- *timeslot-range*—Specifies a value or range of values for time slots that are exempt from suppression during congestion onset. Use a hyphen to indicate a range.

For example, the following command configures the **gsm-abis congestion critical** timeslot range as 1-10:

```
Router(config-if)# gsm-abis congestion critical 1-10
```

Note These are the timeslots that contain signalling and control information exchanged between the BSC and BTS.

Saving Configuration Changes

After you have completed configuring your Cisco RAN Service Module, to prevent the loss of the configuration, you must store the configuration changes by saving it to NVRAM so that the router boots with the configuration you entered.

Step 1 Exit the global configuration mode.

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

Tip You can press **Ctrl-Z** in any mode to return immediately to enable mode (`Router#`), instead of entering **exit**, which returns you to whatever mode you were in previously.

Step 2 Save the configuration changes to NVRAM so that they are not lost during resets, power cycles, or power outages.

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

Monitoring and Managing the Cisco RAN Service Module

You can use Cisco's network management applications, such as Cisco Mobile Wireless Transport Manager (MWTM), to monitor and manage the Cisco RAN Service Module. This Network Management tool provides monitoring and management capabilities to the RAN-O solution. The Cisco MWTM addresses the element-management requirements of mobile operators and provides fault, configuration, and troubleshooting capability. The Cisco MWTM provides the following key features:

- Event Monitoring
- Web-Based Reporting
- Auto Discovery and Topology
- Inventory
- OSS Integration
- Security
- Client/Server Architecture
- Multiple OS Support

The Cisco MWTM integrates with any SNMP-based monitoring system, such as Cisco Info Center products. In addition, the Cisco MWTM collects a large amount of performance data that can be exported or directly accessed from the database. This data can then be used by performance reporting applications.

Additional information can be found in the following publications of the Cisco MWTM documentation set:

- *Cisco Mobile Wireless Transport Manager User Guide*
- *Cisco Mobile Wireless Transport Manager Release Notes*
- *Cisco Mobile Wireless Transport Manager Online Help System*

Enabling the RAN Service Module for Remote Network Management

To enable remote network management of the Cisco RAN Service Module, do the following:

Step 1 At the privileged EXEC prompt, enter the following command to access the configuration mode:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Step 2 At the configuration prompt, enter the following command to assign a host name to each of the network management workstations:

```
Router(config)# ip host hostname ip_address
```

Where *hostname* is the name assigned to the Operations and Maintenance (O&M) workstation and *ip_address* is the address of the network management workstation.

Step 3 Enter the following commands to create a loopback interface for O&M (see the "Configuring Gigabit Ethernet Interfaces" section on page 4-5 for more information):

```
Router(config)# interface loopback number
Router(config-if)# ip address ip_address subnet_mask
```

Step 4 Exit interface configuration mode:

```
Router(config-if)# exit
```

Step 5 At the configuration prompt, enter the following command to specify the recipient of a Simple Network Management Protocol (SNMP) notification operation:

```
Router(config)# snmp-server host hostname [traps | informs] [version {1 | 2c | 3 [auth |
noauth | priv]}] community-string [udp-port port] [notification-type]
```

Where *hostname* is the name assigned to the Cisco Info Center workstation with the **ip host** command in Step 2.

Note See the "Configuring for SNMP Support" section on page 4-30 for more information about configuring Steps 5 through 8 in this procedure.

Step 6 Enter the following commands to specify the public and private SNMP community names:

```
Router(config)# snmp-server community public RO
Router(config)# snmp-server community private RW
```

Step 7 Enter the following command to enable the sending of SNMP traps:

```
Router(config)# snmp-server enable traps
```

Step 8 Enter the following command to specify the loopback interface from which SNMP traps should originate:

```
Router(config)# snmp-server trap-source loopback number
```

Where *number* is the number of the loopback interface you configured for the O&M in Step 3.

Step 9 At the configuration prompt, press Ctrl-Z to exit configuration mode.

Step 10 Write the new configuration to nonvolatile memory as follows:

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

Show Commands for Monitoring the Cisco RAN Service Module

To monitor and maintain the Cisco RAN Service Module, use the following commands:

Command	Purpose
show controllers	Displays all CPU controllers.
show controllers gigabit ethernet *cpu/port*	Displays information about initialization block, transmit ring, receive ring and errors for the Fast Ethernet controller chip.
show controllers e1	Displays information about the controller status specific to the controller hardware. It also displays statistics about the E1 link. If you specify a CPU and port number, statistics for each 15 minute period will be displayed.
show controllers t1	Displays information about the T1 controllers.
show gsm-abis efficiency [history]	Displays the history of the GSM efficiency averages for compression/decompression at 1-second, 5-second, 1-minute, 5-minute, and 1-hour intervals.
show gsm-abis errors	Displays error statistics counters of the GSM for compression/decompression.
show gsm-abis packets	Displays packet statistics counters of the GSM for compression/decompression.
show gsm-abis peering [details | brief]	Displays peering status, statistics, and history of the GSM compression/decompression.
show interface *type cpu/port:channel*	Displays the configuration and status of the specified interface.
show interface gigabit ethernet *cpu/port*	Displays the status of the Gigabit Ethernet (GigE) interface.
show ip rtp header-compression	Displays RTP header compression statistics.
show ppp multilink	Displays MLP and multilink bundle information.
show ppp multilink interface *number*	Displays multilink information for the specified interface.
show protocols	Displays the protocols configured for the router and the individual interfaces.
show umts congestion [**atm]**	Displays the UMTS Congestion state.
show umts-iub efficiency	Displays the history of the UMTS Iub interface efficiency averages for compression/decompression at 1-second, 5-second, 1-minute, 5-minute, and 1-hour intervals.

Command	Purpose
show umts-iub errors	Displays error statistics UMTS-Iub interface.
show umts-iub packets	Displays packet statistics of the UMTS-Iub interface.
show umts-iub peering [details | brief]	Displays peering status, statistics, and history of the UMTS Iub interface.
show umts-iub pvc	Displays the pvc mapping of the UMTS Iub interface
show umts-profile	Displays how the profile is defined and which interfaces are applied.
show controller vc4	Displays the status for the VC4 since some of the line information may be independent of any individual ATM interface.
show controller atm x/y	Displays the controller information for an atm controller.
show provisioned config	Displays the E1T1 controllers that have been provisioned with port configurations.

Where to Go Next

At this point you can proceed to the following:

- The Cisco IOS software configuration guide and command reference publications for more advanced configuration topics. These publications are available on the Documentation DVD that came with your router, available online at Cisco.com, or you can order printed copies.
- The *System Error Messages* and *Debug Command Reference* publications for troubleshooting information available online at Cisco.com.