CHAPTER 18

Configuring MLPPP Backhaul

To configure an MLPPP backhaul, complete the following tasks:

Configuring the Card Type

Perform a basic card type configuration by enabling the router, enabling an interface, and specifying the card type as described below. 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 select and configure a card type, follow these steps:

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 3	**card type {e1 | t1}** *slot subslot* **Example:** `Router(config)# card type e1 0 1`	Sets the card type. The command has the following syntax: • *slot*—Slot number of the interface. • *subslot*—VWIC slot number. The example shows how to configure a T1/E HWIC in the first HWIC slot as an E1 card. When the command is used for the first time, the configuration takes effect immediately. A subsequent change in the card type does not take effect unless you enter the **reload** command or reboot the router. **Note** When you are using the **card type** command to change the configuration of an installed card, you must first enter the **no card type** {**e1** | **t1**} *slot subslot* command. Then enter the **card type** {**e1** | **t1**} *slot subslot* command for the new configuration information.

Step 4	**exit**
	
	Example:
	`Router(config)# exit`
	`Router#`
		Exit configuration mode.

Configuring E1 Controllers

Perform a basic E1 controller configuration by specifying the E1 controller, entering the clock source, 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 in global configuration mode:

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command	Purpose

Step 3	**controller e1** *slot/port*
	
	Example:
	`Router(config)# controller e1 0/0`
	`Router(config-controller)#`
		Specifies the controller that you want to configure. Controller E1 0/0 maps to the T1/E1 HWIC card in HWIC slot 0.
		
		The example shows how to specify the E1 controller as the first port of the T1/E1 HWIC card in slot 0.

Step 4	**framing {crc4 | no-crc4}** **Example:** `Router(config-controller)# framing crc4`	Specifies the framing type.
Step 5	**linecode {ami | hdb3}** **Example:** `Router(config-controller)# linecode ami`	Specifies the line code format.
Step 6	**mode {atm | cas}** **Example:** `Router(config-controller)# mode cas`	Sets the controller in ATM or channel-associated signaling (CAS) mode.
Step 7	**clock source {line | internal} [bits]** **Example:** `Router(config-controller)# clock source line`	Specifies the clocking source. The syntax is: • *line*—Specifies the E1 line from which the clocking is taken. • *internal*—Internal clocking. • bits—Building Integrated Timing Supply (BITS) clocking. The example shows how to configure the clock source for the E1 controller. **Note** When you are using the **clock source** command to change the configuration of an installed card, you must enter the **no clock source** command first. Then enter the **clock source** command for the new configuration information.

	Command	Purpose
Step 8	`Router(config-controller)#` **`channel-group`** *`channel-no`* **`timeslots`** *`timeslot-list`* **`speed {64}`** **Example:** `Router(config-controller)#` **`channel-group 0 timeslots 1-31 speed 64`**	Specifies the channel-group and time slots to be mapped. After you configure a channel-group, the serial interface is automatically created. The syntax is: • *channel-no*—ID number to identify the channel group. The valid range is from 0–30. • *timeslot-list*—Timeslots (DS0s) to include in this channel-group. The valid time slots are from 1–31. • **speed {64}**—The speed of the DS0. The example configures the channel-group and time slots for the E1 controller: **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 9	`Router(config-controller)#` **`exit`** `Router(config)#`	Exits controller configuration mode.
Step 10	**`interface serial`** *`slot/port`***`:channel`** **Example:** `Router(config)#` **`interface serial 0/0:1`** `Router(config-if)#`	Configures the serial interface. Specify the E1 slot, port number, and channel-group. When the prompt changes to `Router(config-if)`, you have entered interface configuration mode. **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 11	`Router(config-if)#` **`encapsulation ppp`**	Specifies PPP encapsulation on the interface.
Step 12	`keepalive [period [retries]]` **Example:** `Router(config-if)#` **`keepalive`** [*`period`* [*`retries`*]]	Enables keepalive packets on the interface and specify the number of times keepalive packets are sent without a response before the router disables the interface.
Step 13	`Router(config-if)#` **`end`** `Router#`	Exits interface configuration mode.

Configuring T1 Controllers

Use the following steps 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 in the global configuration mode:

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**card type {e1 | t1}** *slot subslot* **Example:** `Router(config)# card type t1 0 1`	Sets the card type. The command has the following syntax: • *slot*—Slot number of the interface. • *subslot*—The VWIC slot number. Controller T1 0/0 maps to the T1/E1 HWIC card in HWIC slot 0. The example shows how to configure a T1/E HWIC in the first HWIC slot as an T1 card. When the command is used for the first time, the configuration takes effect immediately. A subsequent change in the card type does not take effect unless you enter the **reload** command or reboot the router. **Note** When you are using the **card type** command to change the configuration of an installed card, you must first enter the **no card type** {**e1** | **t1**} slot subslot command. Then enter the **card type** {**e1** | **t1**} *slot subslot* command for the new configuration information.
Step 4	`Router(config-controller)#` **framing esf**	Specifies the framing type.
Step 5	`Router(config-controller)#` **linecode b8zs**	Specifies the line code format.
Step 6	`Router(config-controller)#` **mode** {atm | cas}	Set the controller in ATM or channel-associated signaling (CAS) mode.
Step 7	`Router(config-controller)#` **channel-group 0 timeslots 1-24 speed 56**	Specifies the channel-group and time slots to be mapped. After you configure a channel-group, the serial interface is automatically created. **Note** The default speed of the channel-group is 56.
Step 8	`Router(config-controller)#` **cablelength {long [-15db | -22.5db | -7.5db | 0db] short [110ft | 220ft | 330ft| 440ft | 550ft | 600ft]}**	Configures the cable length.
Step 9	`Router(config-controller)#` **exit**	Exits controller configuration mode.
Step 10	`Router(config)#` **interface serial** *slot/port*:**channel**	Configures the serial interface. Specify the T1 slot (always 0), port number, and channel-group.

	Command	Purpose
Step 11	`Router(config-if)# encapsulation ppp`	Enters the following command to configure PPP encapsulation.
Step 12	`Router(config-if)# keepalive [period [retries]]`	Enables keepalive packets on the interface and specify the number of times that keepalive packets will be sent without a response the interface is brought down:

Step 13	**exit**
	
	Example:
	`Router(config)# exit`
	`Router#`
		Exits configuration mode.

Configuring ATM IMA

Inverse multiplexing provides the capability to transmit and receive a single high-speed data stream over multiple slower-speed physical links. In Inverse Multiplexing over ATM (IMA), the originating stream of ATM cells is divided so that complete ATM cells are transmitted in round-robin order across the set of ATM links. Follow these steps to configure ATM IMA on the Cisco MWR 2941.

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# card type e1 0 0`	Specifies the slot and port number of the E1 or T1 interface.
Step 4	`Router(config)# controller E1 0/4` `Router(config-controller)#`	Specifies the controller interface on which you want to enable IMA.
Step 5	`Router(config-controller)# clock source internal`	Set sthe clock source to internal.
Step 6	`Router(config-controller)# ima-group 0 scrambling-payload`	Assigns the interface to an IMA group, and set the scrambling-payload parameter to randomize the ATM cell payload frames. This command assigns the interface to IMA group 0. **Note** This command automatically creates an ATM0/IMAx interface.
Step 7		To add another member link, repeat Step 3 to Step 6.
Step 8	`Router(config-controller)# exit` `Router(config)#`	Exits the controller interface.

	Command	Purpose
Step 9	**interface ATM***slot*/**IMA**<*group-number*> **Example:** `Router(config-if)# interface atm0/ima0`	Specify the slot location and port of IMA interface group. • *slot*—The slot location of the ATM IMA port adapter. • *group-number*—The group number of the IMA group. The example specifies the slot number as 0 and the group number as 0. **Note** To explicitly configure the IMA group ID for the IMA interface, you may use the optional **ima group-id** command. You cannot configure the same IMA group ID on two different IMA interfaces; therefore, if you configure an IMA group ID with the system-selected default ID already configured on an IMA interface, the system toggles the IMA interface to make the user-configured IMA group ID the effective IMA group ID. At the same, the system toggles the original IMA interface to select a different IMA group ID.
Step 10	`Router(config-if)# no ip address`	Disables the IP address configuration for the physical layer interface.
Step 11	`Router(config-if)# atm bandwidth dynamic`	Specifies the ATM bandwidth as dynamic.
Step 12	`Router(config-if)# no atm ilmi-keepalive`	Disables the Interim Local Management Interface (ILMI) keepalive parameters.

Step 13	**exit**
	
	Example:
	`Router(config)# exit`
	`Router#`
		Exits configuration mode.

Note The above configuration has one IMA shorthaul with two member links (atm0/0 and atm0/1).

Configuring T1 and E1 Local Switching

The Cisco MWR 2941 provides support for T1 and E1 local switching. You can use the following commands to configure T1 and E1 local switching:

- **tdm-group**—Configures a list of time slots for creating clear channel groups (pass-through) for time-division multiplexing (TDM) cross-connect.
- **connect**—Defines connections among T1 or E1 controller ports for drop-and-insert (also called TDM cross-connect).

Note Local switching is only supported between onboard T1 and E1 ports; local switching between HWIC T1 and E1 ports is not supported.

Note You cannot add a TDM-GROUP to a controller where the CEM-GROUP is defined. Channel-group also conflicts with CEM-GROUP, which cannot coexist.

Follow these steps to configure T1 and E1 local switching

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**controller** {**t1** | **e1**} *slot/port*	Enter T1 or E1 controller configuration mode.
Step 3	**tdm-group** *tdm-group-no* **timeslot** *timeslot-list*	Specify the TDM group and timeslots for which you want to enable local switching.
Step 4	**controller** {**t1** | **e1**} *slot/port*	Enter T1 or E1 controller configuration mode for the second controller.
Step 5	**tdm-group** *tdm-group-no* **timeslot** *timeslot-list*	Specify the second set TDM group and timeslots for which you want to enable local switching.
Step 6	**exit**	Exit controller configuration mode.
Step 7	**connect** *connection-id* {**t1** | **e1**} *slot/port-1 tdm-group-no-1* {**t1** | **e1**} *slot/port-2 tdm-group-no-2*	Use the connect command to enable local switching.
Step 8	**show connection**	(Optional) Use the **show connection** command to verify your configuration.

For more information about these commands, see the *Cisco MWR 2941 Router Command Reference, Release 15.1(1)MR*.

Configuration Examples for T1 and E1 Local Switching

The following examples show how to use T1 and E1 local switching:

- TDM Local Switching—E1
- TDM Local Switching—T1
- Non-Channelized Local Switching
- Channelized Local Switching
- Channelized Local Switching on Multiple Channels
- Channelized Local Switching with Segmented Timeslots

TDM Local Switching—E1

```
controller E1 0/0
 tdm-group 0 timeslots 1-31
controller E1 0/1
 tdm-group 0 timeslots 1-31
connect st E1 0/0 0 E1 0/1 0
```

TDM Local Switching—T1

```
controller T1 0/0
 tdm-group 0 timeslots 1-24
controller T1 0/1
 tdm-group 0 timeslots 1-24
connect stanley T1 0/0 0 T1 0/1 0
```

Non-Channelized Local Switching

```
controller E1 0/0
 tdm-group 0 timeslots 1-31

controller E1 0/1
 tdm-group 0 timeslots 1-31

connect st1 E1 0/0 0 E1 0/1 0
```

Channelized Local Switching

```
controller E1 0/0
 tdm-group 0 timeslots 1-10
 tdm-group 1 timeslots 11-20

controller E1 0/1
tdm-group 0 timeslots 1-10
 tdm-group 1 timeslots 11-20

connect st1 E1 0/0 0 E1 0/1 0
connect st2 E1 0/0 1 E1 0/1 1
```

Channelized Local Switching on Multiple Channels

```
controller E1 0/0
 tdm-group 0 timeslots 1-10
 tdm-group 1 timeslots 11-20
tdm-group 2 timeslots 21-25
tdm-group 3 timeslots 26-31
controller E1 0/1
tdm-group 0 timeslots 1-10
 tdm-group 1 timeslots 11-20
tdm-group 2 timeslots 21-25
tdm-group 3 timeslots 26-31

connect st1 E1 0/0 0 E1 0/1 0
connect st2 E1 0/0 1 E1 0/1 1
connect st3 E1 0/0 2 E1 0/1 2
connect st4 E1 0/0 3 E1 0/1 3
```

Channelized Local Switching with Segmented Timeslots

```
controller E1 0/0
 tdm-group 0 timeslots 1,3,20-22
 tdm-group 1 timeslots 24,26,29-30
controller E1 0/1
tdm-group 0 timeslots 1,3,20-22
 tdm-group 1 timeslots 24,26,29-30

connect st1 E1 0/0 0 E1 0/1 0
connect st2 E1 0/0 1 E1 0/1 1
```

Configuring a Multilink Backhaul Interface

A multilink interface is a 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 MWR 2941 router can support up to 16 E1/T1 connections through the multilink interface, ranging from 12 bundles of 1 E1/T1 each to a single bundle containing 16 E1/T1 bundles.

Complete the following tasks to configure a multilink backhaul interface.

- Creating a Multilink Bundle
- Configuring PFC and ACFC, page 18-11
- Enabling Multilink and Identifying the Multilink Interface, page 18-13
- Enabling Real-Time Transport Protocol (RTP) Header Compression On a Multilink Inteface, page 18-15
- Enabling Distributed IPHC Offload on a Multilink Interface, page 18-15

Creating a Multilink Bundle

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

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	`Router(config)# interface multilink` *group-number*
	
	Example:
	`Router(config)# interface multilink5`
	`Router(config-if)#`
		Creates a multilink bundle and enter the interface configuration mode:
		• *group-number*—Number of the multilink bundle.
		The example creates a multilink bundle 5.
		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.

	Command	Purpose

Step 4	`Router(config-if)# ip address address [subnet mask]`
	
	Example:
	`Router(config-if)# ip address 10.10.10.2 255.255.255.0`
		Assigns an IP address to the multilink interface.
		• *address*— IP address.
		• *subnet mask*—Network mask of IP address.
		The example configures an IP address and subnet mask.

Step 5	**exit**
	
	Example:
	`Router(config)# exit`
	`Router#`
		Exits configuration mode.

Configuring PFC and ACFC

Protocol-Field-Compression (PFC) and Address-and-Control-Field-Compression (AFC) are PPP compression methods defined in RFCs 1661 and 1662. PFC allows for compression of the PPP Protocol field; ACFC allows for compression of the PPP Data Link Layer Address and Control fields.

Follow these steps to configure PFC and ACFC handling during PPP negotiation to be configured. By default, PFC/ACFC handling is not enabled.

Note The recommended PFC and ACFC handling in the Cisco MWR 2941 router is: **acfc local request, acfc remote apply, pfc local request, and pfc remote apply**.

Configuring PFC

To configure PFC handling during PPP negotiation, follow these steps:

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	`Router(config-if)# ppp pfc local {request | forbid}`
	
	Example:
	`Router(config-if)# ppp pfc local request`
		Configures how the router handles PFC in its outbound configuration requests, use the **ppp pfc local** command. The syntax is as follows:
		• **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.
		The example shows how to create a method for the router to manage PFC.

	Command	Purpose

Step 4	`Router(config-if)# ppp pfc remote {apply | reject | ignore}`
	
	Example:
	`Router(config)# ppp pfc remote apply`
		Specifies how the router manages the PFC option in configuration requests received from a remote peer. The syntax is as follows:
		• **apply**—Specifies that PFC options are accepted and ACFC may be performed on frames sent to the remote peer.
		• **reject**—Specifies that PFC options are explicitly ignored.
		• **ignore**—Specifies that PFC options are accepted, but ACFC is not performed on frames sent to the remote peer.
		The example shows how to allow PFC options to be accepted.

Step 5	**exit**
	
	Example:
	`Router(config)# exit`
	`Router#`
		Exits configuration mode.

Configuring ACFC

To configure ACFC handling during PPP negotiation, follow these steps, while in interface configuration mode:

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	`Router(config-if)# ppp acfc local {request | forbid}`
	
	Example:
	`Router(config-if)# ppp acfc local request`
		Specifies how the router handles ACFC in outbound configuration requests. The syntax is as follows:
		• **request**—Specifies that the ACFC option is included in outbound configuration requests.
		• **forbid**—Specifies that the ACFC option is not sent in outbound configuration requests, and requests from a remote peer to add the ACFC option are not accepted.

	Command	Purpose

Step 4	`Router(config-if)# ppp acfc remote {apply | reject | ignore}`
	
	Example:
	`Router(config-if)# ppp acfc remote apply`
		Specifies how the router handles the ACFC option in configuration requests received from a remote peer. The syntax is as follows:
		• **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.
		The example allows ACFC options to be accepted.

Step 5	**exit**
	
	Example:
	`Router(config)# exit`
	`Router#`
		Exit configuration mode.

Enabling Multilink and Identifying the Multilink Interface

To enable multilink and identify the multilink interface, follow these steps, while in interface configuration mode:

Note If you modify parameters for an MLPPP bundle while it is active, the changes do not take effect until the Cisco MWR 2941 renegotiates the bundle connection.

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config-if)# ppp multilink`	Enables multilink PPP operation.
Step 4	`Router(config-if)# ppp multilink group` *group-number* **Example:** `Router(config-if)# ppp multilink group 5`	Configures the identification number for the multilink interface. The syntax is as follows: • ***group-number***—Multilink group number. The example restricts (identifies) the multilink interface that can be negotiated to multilink interface 5.

	Command	Purpose

Step 5	`Router(config-if)# keepalive [period [retries]]`
	
	Example:
	`Router(config-if)# keepalive 1 5`
		Enables keepalive packets on the interface and specifies the number of times the keepalive packets are sent without a response before the router disables the interface. The syntax is as follows:
		• ***period***—(Optional) Integer value in seconds greater than 0. The default is 10.
		• ***retries***—(Optional) Specifies the number of times that the device will continue to send keepalive packets without response before bringing the interface down. Integer value greater than 1 and less than 255. If omitted, the value that was previously set is used; if no value was specified previously, the default of 5 is used.

Step 6	**exit**
	
	Example:
	`Router(config)# exit`
	`Router#`
		Exits configuration mode.

Configuring Additional MLPPP Settings

You can perform a variety of other configurations on an MLPPP bundle, including the following:

- Modifying the maximum fragment size
- Modifying fragmentation settings
- Enabling or disabling fragmentation
- Enabling or disabling interleaving
- Configuring distributed MLPPP (dMLPPP)
- Configuring multiclass MLPPP

For more information about configuring MLPPP, see the Dial Configuration Guide, Cisco IOS Release 15.0S.

MLPPP Offload

By default, the Cisco MWR 2941 offloads processing for distributed MLPPP (dMLPPP) to the network processor for improved performance. However, the Cisco MWR 2941 does not support some dMLPPP settings on offloaded bundles. The Cisco MWR 2941 does not support the following options on offloaded dMLPPP bundles:

- **ppp multilink idle-link**
- **ppp multilink queue depth**
- **ppp multilink fragment maximum**
- **ppp multilink slippage**
- **ppp timeout multilink lost-fragment**

Note If you have a bundle that requires the use of these options, contact Cisco support for assistance.

Enabling Real-Time Transport Protocol (RTP) Header Compression On a Multilink Inteface

To enable RTP header compression, follow these steps while in the interface configuration mode:

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *number* **Example:** `Router# configure interface multilink 1`	Enters multilink interface configuration mode to configure the multilink bundle.
Step 4	`Router(config-if)#` **ip rtp header-compression** [**ietf-format**] [**periodic-refresh**] **Example:** `Router(config-if)#` **ip rtp header-compression ietf-format periodic-refresh**	Enable RTP header-compression using the **ip rtp header-compression** command. The syntax is as follows: • **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. The example enables RTP header-compression in the Internet Engineering Task Force (IETF) format by suppressing the IP ID in the RTP/UDP header compression. **Note** IP header compression is only supported when MLPPP operates on the host processor; it is not supported when MLPPP is offloaded.
Step 5	**exit** **Example:** `Router(config)#` **exit** `Router#`	Exit configuration mode.

Enabling Distributed IPHC Offload on a Multilink Interface

Additionally, you can configure distributed IPHC (dIPHC) on a multilink interface for improved Multilink Point-to-Point Protocol (MLPPP) performance.

Distributed IP Header Compression (dIPHC) allows the Cisco MWR 2941 to compress IP packet headers for more efficient use of bandwidth.

dIPHC Usage Notes

- The Cisco MWR 2941 supports dIPHC for UDP traffic only.
- The Cisco MWR 2941 supports up to 300 IPHC sessions over 3 multilinks .
- The Cisco MWR 2941 supports only the IEFT format of IPHC.
- The Cisco MWR 2941 supports dIPHC only for UDP traffic.
- After enabling dIPHC on an interface, ensure that you "flap" the interface by using the **shutdown** and **no shutdown** commands in interface configuration mode for the configuration to take effect.
- If a QoS policy is attached to the multilink interface on which IPHC has been enable, ensure that you configure the **platform header-compression match access-group** command in interface configuration mode to apply an extended named Access Control List (ACL) of static flows (source IP address, destination IP address, and destination port) to compress using IPHC.

  You must configure the matching ACL as an extended named access-list and add the static flow configuration to a Multilink interface that is in an UP state and on which the **ip rtp header-compression** command is configured.

Note If you modify parameters for an MLPPP bundle while it is active, the changes do not take effect until the Cisco MWR 2941 renegotiates the bundle connection.

To enable dIPHC offload on an multilink interface, complete the following task:

	Command	Purpose
Step 6	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 7	**interface multilink** *number* **Example:** `Router# configure interface multilink 1`	Enters multilink interface configuration mode to configure the multilink bundle.
Step 8	**ip rtp header-compression ietf-format** **Example:** `Router(config-if)# ip rtp header-compression ietf-format`	Enables IEFT RTP header compression. **Note** The Cisco MWR 2941 support only IEFT RTP header compression.
Step 9	**shutdown** **Example:** `Router(config-if)# shutdown`	Disables the interface.
Step 10	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.

	Command	Purpose

Step 11	**exit**
	
	Example:
	`Router(config-if)# exit`
	`Router(config)#`
		Exits interface configuration mode.

Step 12	**exit**
	
	Example:
	`Router(config)# exit`
	`Router#`
		Exits global configuration mode.

Step 13	**show ip rtp header-compression multilink** **Example:** `Router# show ip rtp header-compression`	Displays Real-Time Transport Protocol (RTP) header compression statistics on the multilink interface.
Step 14	**show platform hardware winpath iphc** **Example:** `Router# show platform hardware winpath iphc`	Displays the IPHC information for the Winpath hardware device on the Cisco MWR 2941.