Time Division Multiplexing Configuration Guide, Cisco IOS XE 17 (Cisco ASR 900 Series)

First Published: 2023-10-19

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

CONTENTS

CHAPTER 1

Configuring Pseudowire

This chapter provides information about configuring pseudowire (PW) features on the router.

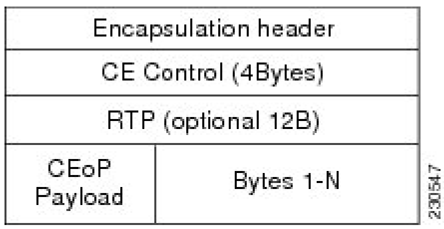
Structure-Agnostic TDM over Packet

SAToP encapsulates time division multiplexing (TDM) bit-streams (T1, E1, T3, E3) as PWs over public switched networks. It disregards any structure that may be imposed on streams, in particular the structure imposed by the standard TDM framing.

The protocol used for emulation of these services does not depend on the method in which attachment circuits are delivered to the provider edge (PE) devices. For example, a T1 attachment circuit is treated the same way for all delivery methods, including copper, multiplex in a T3 circuit, a virtual tributary of a SONET/SDH circuit, or unstructured Circuit Emulation Service (CES).

In SAToP mode the interface is considered as a continuous framed bit stream. The packetization of the stream is done according to IETF RFC 4553. All signaling is carried out transparently as a part of a bit stream. Figure 1: Unstructured SAToP Mode Frame Format, on page 2 shows the frame format in Unstructured SAToP mode.

Figure 1: Unstructured SAToP Mode Frame Format



#unique_3 unique_3_Connect_42_tab_1729930 shows the payload and jitter limits for the T1 lines in the SAToP frame format.

Table 1: SAToP T1 Frame: Payload and Jitter Limits

Maximum Payload	Maximum Jitter	Minimum Jitter	Minimum Payload	Maximum Jitter	Minimum Jitter
960	320	10	192	64	2

#unique_3 unique_3_Connect_42_tab_1729963 shows the payload and jitter limits for the E1 lines in the SAToP frame format.

Table 2: SAToP E1 Frame: Payload and Jitter Limits

Maximum Payload	Maximum Jitter	Minimum Jitter	Minimum Payload	Maximum Jitter	Minimum Jitter
1280	320	10	256	64	2

For instructions on how to configure SAToP, see Configuring Structure-Agnostic TDM over Packet (SAToP), on page 19.

Circuit Emulation Overview

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

The Cisco ASR 903 Series Router supports two pseudowire types that utilize CEM transport: Structure-Agnostic TDM over Packet (SAToP) and Circuit Emulation Service over Packet-Switched Network (CESoPSN). The following sections provide an overview of these pseudowire types.

Starting with Cisco IOS XE Release 3.15, the 32xT1E1 and 8x T1/E1 interface modules support CEM CESoP and SATOP configurations with fractional timeslots.

With the 32xT1/E1 and 8xT1/E1 interface modules, the channelized CEM circuits configured under a single port (fractional timeslot) cannot be deleted or modified, unless the circuits created after the first CEM circuits are deleted or modified.

The following CEM circuits are supported on the 32xT1/E1 interface module:

T1 mode

- 192 CESOP circuits with fractional timeslot
- 32 CESOP circuit full timeslot
- 32 SATOP circuits

E1 mode

- 256 CESOP circuit with fractional timeslot
- 32 CESOP circuit full timeslot
- 32 SATOP circuit

Note CEM pseudowire with local loopback at the CEM sides of PEs results in propagating AIS and L-bit alarms.

The L-bit packets are dropped for the following interface modules:

- A900-IMA8D
- A900-IMA16D
- A900-IMA32D
- A900-IMA4OS

Circuit Emulation Service over Packet-Switched Network

CESoPSN encapsulates structured TDM signals as PWs over public switched networks (PSNs). It complements similar work for structure-agnostic emulation of TDM bit streams, such as SAToP. Emulation of circuits saves PSN bandwidth and supports DS0-level grooming and distributed cross-connect applications. It also enhances resilience of CE devices due to the effects of loss of packets in the PSN.

CESoPSN identifies framing and sends only the payload, which can either be channelized T1s within DS3 or DS0s within T1. DS0s can be bundled to the same packet. The CESoPSN mode is based on IETF RFC 5086.

Each supported interface can be configured individually to any supported mode. The supported services comply with IETF and ITU drafts and standards.

Figure 2: Structured CESoPSN Mode Frame Format, on page 4 shows the frame format in CESoPSN mode.

Figure 2: Structured CESoPSN Mode Frame Format

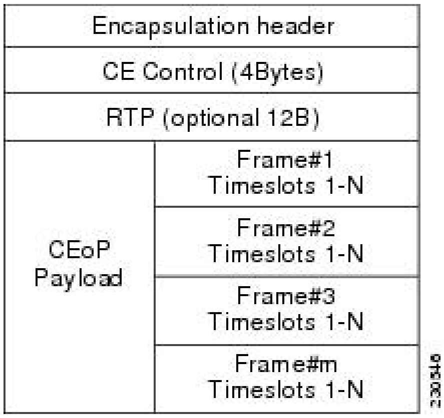


Table 3: CESoPSN DS0 Lines: Payload and Jitter Limits, on page 4 shows the payload and jitter for the DS0 lines in the CESoPSN mode.

Table 3: CESoPSN DS0 Lines: Payload and Jitter Limits

[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]

[illegible]

For instructions on how to configure SAToP, see Configuring Structure-Agnostic TDM over Packet (SAToP), on page 19.

Asynchronous Transfer Mode over MPLS

An ATM over MPLS (AToM) PW is used to carry Asynchronous Transfer Mode (ATM) cells over an MPLS network. It is an evolutionary technology that allows you to migrate packet networks from legacy networks, while providing transport for legacy applications. AToM is particularly useful for transporting 3G voice traffic over MPLS networks.

You can configure AToM in the following modes:

- N-to-1 Cell—Maps one or more ATM virtual channel connections (VCCs) or virtual permanent connection (VPCs) to a single pseudowire.
- 1-to-1 Cell—Maps a single ATM VCC or VPC to a single pseudowire.
- Port—Maps a single physical port to a single pseudowire connection.

The Cisco ASR 903 Series Router also supports cell packing and PVC mapping for AToM pseudowires.

Note This release does not support AToM N-to-1 Cell Mode or 1-to-1 Cell Mode.

For more information about how to configure AToM, see Configuring an ATM over MPLS Pseudowire, on page 23.

Transportation of Service Using Ethernet over MPLS

Ethernet over MPLS (EoMPLS) PWs provide a tunneling mechanism for Ethernet traffic through an MPLS-enabled Layer 3 core network. EoMPLS PWs encapsulate Ethernet protocol data units (PDUs) inside MPLS packets and use label switching to forward them across an MPLS network. EoMPLS PWs are an evolutionary technology that allows you to migrate packet networks from legacy networks while providing transport for legacy applications. EoMPLS PWs also simplify provisioning, since the provider edge equipment only requires Layer 2 connectivity to the connected customer edge (CE) equipment. The Cisco ASR 903 Series Router implementation of EoMPLS PWs is compliant with the RFC 4447 and 4448 standards.

The Cisco ASR 903 Series Router supports VLAN rewriting on EoMPLS PWs. If the two networks use different VLAN IDs, the router rewrites PW packets using the appropriate VLAN number for the local network.

For instructions on how to create an EoMPLS PW, see Configuring an Ethernet over MPLS Pseudowire, on page 32.

Limitations

If you are running Cisco IOS XE Release 3.17S and later releases, the following limitations apply:

- Channel associated signaling (CAS) is not supported on the T1/E1 and OC-3 interface modules on the router.
- BGP PIC is not supported for MPLS/LDP over MLPPP and POS in the core.
- BGP PIC is not supported for Multi-segment Pseudowire or Pseudowire switching.
- BGP PIC is not supported for VPLS and H-VPLS.
- BGP PIC is not supported for IPv6.
- If BGP PIC is enabled, Multi-hop BFD should not be configured using the **bfd neighbor fall-over**r **bfd** command.
- If BGP PIC is enabled, **neighbor** *ip-address* **weight** *weight* command should not be configured.
- If BGP PIC is enabled, **bgp nexthop trigger delay 6** under the **address-family ipv4** command and **bgp nexthop trigger delay 7** under the **address-family vpnv4** command should be configured. For information on the configuration examples for BGP PIC–TDM, see Example: BGP PIC with TDM-PW Configuration, on page 41.
- If BGP PIC is enabled and the targeted LDP for VPWS cross-connect services are established over BGP, perform the following tasks:
  - configure Pseudowire-class (pw-class) with encapsulation "mpls"

- configure **no status control-plane route-watch** under the pw-class
- associate the pw-class with the VPWS cross-connect configurations

If you are running Cisco IOS-XE 3.18S, the following restrictions apply for BGP PIC with MPLS TE for TDM Pseudowire:

- MPLS TE over MLPPP and POS in the core is not supported.
- Co-existence of BGP PIC with MPLS Traffic Engineering Fast Reroute (MPLS TE FRR) is not supported.

The following restrictions are applicable only if the BFD echo mode is enabled on the Ethernet interface carrying CEM or TDM traffic:

- When the TDM interface module is present in anyone of the slot—0, 1, or 2, then the corresponding Ethernet interface module carrying the CEM traffic should also be present in one of these slots.
- When the TDM interface moduleis present in anyone of the slot—3, 4, or 5, then the corresponding Ethernet interface module carrying the CEM traffic should also be present in one of these slots.

Configuring CEM

This section provides information about how to configure CEM. CEM provides a bridge between a time-division multiplexing (TDM) network and a packet network, such as Multiprotocol Label Switching (MPLS). The router encapsulates the TDM data in the MPLS packets and sends the data over a CEM pseudowire to the remote provider edge (PE) router. Thus, function as a physical communication link across the packet network.

The following sections describe how to configure CEM:

Note Steps for configuring CEM features are also included in the Configuring Structure-Agnostic TDM over Packet (SAToP), on page 19 and Configuring Circuit Emulation Service over Packet-Switched Network (CESoPSN), on page 20 sections.

Configuration Guidelines and Restrictions

- Not all combinations of payload size and dejitter buffer size are supported. If you apply an incompatible payload size or dejitter buffer size configuration, the router rejects it and reverts to the previous configuration.
- We recommend you to tune the dejitter buffer setting across Cisco ASR 900 Series router variants in case of interoperability scenarios to achieve better latency.

Configuring a CEM Group

The following section describes how to configure a CEM group on the Cisco ASR 903 Series Router.

Procedure

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 **controller** {**t1** | e1} *slot/subslot/port*

Example:

```
Router(config)# controller t1 1/0
```

Enters controller configuration mode.

- Use the slot and port arguments to specify the slot number and port number to be configured.

Note The slot number is always 0.

Step 4 **cem-group group-number** {**unframed** | **timeslots** *timeslot*}

Example:

```
Router(config-controller)# cem-group 6 timeslots 1-4,9,10
```

Creates a circuit emulation channel from one or more time slots of a T1 or E1 line.

- The **group-number** keyword identifies the channel number to be used for this channel. For T1 ports, the range is 0 to 23. For E1 ports, the range is 0 to 30.
- Use the **unframed** keyword to specify that a single CEM channel is being created including all time slots and the framing structure of the line.
- Use the **timeslots** keyword and the *timeslot* argument to specify the time slots to be included in the CEM channel. The list of time slots may include commas and hyphens with no spaces between the numbers.

Step 5 **end**

Example:

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

Exits controller configuration mode and returns to privileged EXEC mode.

Using CEM Classes

A CEM class allows you to create a single configuration template for multiple CEM pseudowires. Follow these steps to configure a CEM class:

> **Note** The CEM parameters at the local and remote ends of a CEM circuit must match; otherwise, the pseudowire between the local and remote PE routers will not come up.

> **Note** You cannot apply a CEM class to other pseudowire types such as ATM over MPLS.

Procedure

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 **class cem** *cem-class*

Example:

```
Router(config)# class cem mycemclass
```

Creates a new CEM class

Step 4 **payload-size** *size* | **dejitter-buffer** *buffer-size* | **idle-pattern** *pattern*

Example:

```
Router(config-cem-class)# payload-size 512
```

Example:

```
Router(config-cem-class)# dejitter-buffer 10
```

Example:

```
Router(config-cem-class)# idle-pattern 0x55
```

Enter the configuration commands common to the CEM class. This example specifies a sample rate, payload size, dejitter buffer, and idle pattern.

Step 5 **exit**

Example:

```
Router(config-cem-class)# exit
```

Returns to the config prompt.

Step 6 **interface cem** *slot/subslot*

Example:

Example:

```
Router(config)# interface cem 0/0
```

Example:

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

Example:

```
Router(config-if)# cem 0
```

Example:

```
Router(config-if-cem)# cem class mycemclass
```

Example:

```
Router(config-if-cem)# xconnect 10.10.10.10 200 encapsulation mpls
```

Example:

Configure the CEM interface that you want to use for the new CEM class.

Note The use of the **xconnect** command can vary depending on the type of pseudowire you are configuring.

Step 7 **exit**

Example:

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

Example:

Exits the CEM interface.

Step 8 **exit**

Example:

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

Example:

Exits configuration mode.

Configuring a Clear-Channel ATM Interface

Configuring CEM Parameters

The following sections describe the parameters you can configure for CEM circuits.

Note The CEM parameters at the local and remote ends of a CEM circuit must match; otherwise, the pseudowire between the local and remote PE routers will not come up.

Configuring Payload Size (Optional)

To specify the number of bytes encapsulated into a single IP packet, use the pay-load size command. The size argument specifies the number of bytes in the payload of each packet. The range is from 32 to 1312 bytes.

Default payload sizes for an unstructured CEM channel are as follows:

- E1 = 256 bytes
- T1 = 192 bytes
- DS0 = 32 bytes

Default payload sizes for a structured CEM channel depend on the number of time slots that constitute the channel. Payload size (L in bytes), number of time slots (N), and packetization delay (D in milliseconds) have the following relationship: L = 8*N*D. The default payload size is selected in such a way that the packetization delay is always 1 millisecond. For example, a structured CEM channel of 16xDS0 has a default payload size of 128 bytes.

The payload size must be an integer of the multiple of the number of time slots for structured CEM channels.

Setting the Dejitter Buffer Size

To specify the size of the dejitter buffer used to compensate for the network filter, use the dejitter-buffer size command. The configured dejitter buffer size is converted from milliseconds to packets and rounded up to the next integral number of packets. Use the size argument to specify the size of the buffer, in milliseconds. The range is from 1 to 32 ms; the default is 5 ms.

Setting an Idle Pattern (Optional)

To specify an idle pattern, use the [no] idle-pattern pattern1 command. The payload of each lost CESoPSN data packet must be replaced with the equivalent amount of the replacement data. The range for pattern is from 0x0 to 0xFF; the default idle pattern is 0xFF.

Enabling Dummy Mode

Dummy mode enables a bit pattern for filling in for lost or corrupted frames. To enable dummy mode, use the **dummy-mode** [**last-frame** | **user-defined**] command. The default is last-frame. The following is an example:

```
Router(config-cem)# dummy-mode last-frame
```

Setting a Dummy Pattern

If dummy mode is set to user-defined, you can use the **dummy-pattern** *pattern* command to configure the dummy pattern. The range for *pattern* is from 0x0 to 0xFF. The default dummy pattern is 0xFF. The following is an example:

```
Router(config-cem)# dummy-pattern 0x55
```

Note The dummy-pattern command is *not* supported on the following interface modules:

- 48-Port T3/E3 CEM interface module
- 48-Port T1/E1 CEM interface module
- 1-port OC-192 Interface module or 8-port Low Rate interface module

Shutting Down a CEM Channel

To shut down a CEM channel, use the **shutdown** command in CEM configuration mode. The **shutdown** command is supported only under CEM mode and not under the CEM class.

Configuring CAS

This section provides information about how to configure Channel Associated Signaling (CAS).

Information About CAS

The CAS is a method of signaling, where the signaling information is carried over a signaling resource that is specific to a particular channel. For each channel there is a dedicated and associated signaling channel.

The Cisco ASR Router with RSP2 module supports CAS with 8-port T1/E1 interface modules and is interoperable with 6-port Ear and Mouth (E&M) interface modules.

Note The Cisco ASR Router supports CAS only in the E1 mode for the 8-port T1/E1 interface cards. Use the **card type e1 slot/subslot** command to configure controller in the E1 mode.

In the E1 framing and signaling, each E1 frame supports 32 timeslots or channels. From the available timeslots, the timeslot 17 is used for signaling information and the remaining timeslots are used for voice and data. Hence, this kind of signaling is often referred as CAS.

In the E1 frame, the timeslots are numbered from 1 to 32, where the timeslot 1 is used for frame synchronization and is unavailable for traffic. When the first E1 frame passes through the controller, the first four bits of signaling channel (timeslot 17) are associated with the timeslot 2 and the second four bits are associated with the timeslot 18. In the second E1 frame, the first four bits carry signaling information for the timeslot 3 and the second four bits for the timeslot 19.

Configuring CAS

To configure CAS on the controller interface, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters the global configuration mode.
Step 2	**controller e1** *slot/subslot/port* **Example:** `Router(config)# controller E1 0/4/2`	Enters controller configuration mode to configure the E1 interface. **Note** The CAS is supported only in the El mode. Use the **card type e1 slot/subslot** command to configure controller in the E1 mode.
Step 3	**cas** **Example:** `Router(config-controller)# cas`	Configures CAS on the interface.
Step 4	**clock source internal** **Example:** `Router(config-controller)# clock source internal`	Sets the clocking for individual E1 links.
Step 5	**cem-group** *group-number***timeslots** *time-slot-range* **Example:** `Router(config-controller)# cem-group 0 timeslots 1-31`	Creates a Circuit Emulation Services over Packet Switched Network circuit emulation (CESoPSN) CEM group. • cem-group—Creates a circuit emulation (CEM) channel from one or more time slots of a E1 line.

	Command or Action	Purpose
		• group-number—CEM identifier to be used for this group of time slots. For E1 ports, the range is from 0 to 30. • timeslots—Specifies that a list of time slots is to be used as specified by the time-slot-range argument. • time-slot-range—Specifies the time slots to be included in the CEM channel. The list of time slots may include commas and hyphens with no spaces between the numbers.
Step 6	**end** **Example:** `Router(config-controller)# end`	Exits the controller session and returns to the configuration mode.

What to do next

You can configure CEM interface and parameters such as xconnect.

Verifying CAS Configuration

Use the **show cem circuit** *cem-group-id* command to display CEM statistics for the configured CEM circuits. If xconnect is configured under the circuit, the command output also includes information about the attached circuit.

Following is a sample output of the **show cem circuit** command to display the detailed information about CEM circuits configured on the router:

```
Router# show cem circuit 0
CEM0/3/0, ID: 0, Line: UP, Admin: UP, Ckt: ACTIVE
Controller state: up, T1/E1 state: up
Idle Pattern: 0xFF, Idle CAS: 0x8
Dejitter: 8 (In use: 0)
Payload Size: 32
Framing: Framed (DS0 channels: 1)
CEM Defects Set
None

Signalling: No CAS
RTP: No RTP

Ingress Pkts:    5001                   Dropped:             0
Egress Pkts:     5001                   Dropped:             0

CEM Counter Details
Input Errors:    0                    Output Errors:       0
Pkts Missing:    0                    Pkts Reordered:      0
Misorder Drops:  0                    JitterBuf Underrun:  0
```

```
Error Sec:        0                     Severly Errored Sec: 0
Unavailable Sec: 0                      Failure Counts:      0
Pkts Malformed:  0                      JitterBuf Overrun:   0
```

Note The **show cem circuit** command displays No CAS for the **Signaling** field. The No CAS is displayed since CAS is not enabled at the CEM interface level. The CAS is enabled for the entire port and you cannot enable or disable CAS at the CEM level. To view the CAS configuration, use the **show running-config** command.

Configuration Examples for CAS

The following example shows how to configure CAS on a CEM interface on the router:

```
Router# configure terminal
Router(config)# controller E1 0/4/2
Router(config-controller)# cas
Router(config-controller)# clock source internal
Router(config-controller)# cem-group 0 timeslots 1
Router(config-controller)# exit
```

Configuring ATM

The following sections describe how to configure ATM features on the T1/E1 interface module:

Configuring a Clear-Channel ATM Interface

To configure the T1 interface module for clear-channel ATM, follow these steps:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**controller {t1}** *slot/subslot/port* **Example:** `Router(config)# controller t1 0/3/0`	Selects the T1 controller for the port you are configuring (where *slot* /*subslot* identifies the location and /*port* identifies the port).

	Command or Action	Purpose
Step 4	**atm** **Example:** `Router(config-controller)# atm`	Configures the port (interface) for clear-channel ATM. The router creates an ATM interface whose format is atm/*slot* /*subslot* /*port* . **Note** The slot number is always 0.
Step 5	**end** **Example:** `Router(config-controller)# end`	Exits configuration mode.

What to do next

To access the new ATM interface, use the **interface atm***slot/subslot/port* command.

This configuration creates an ATM interface that you can use for a clear-channel pseudowire and other features. For more information about configuring pseudowires, see Configuring Pseudowire, on page 1

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 ASR 903 Series Router.

Note ATM IMA is used as an element in configuring ATM over MPLS pseudowires. For more information about configuring pseudowires, see Configuring Pseudowire, on page 1

Note The maximum ATM over MPLS pseudowires supported per T1/E1 interface module is 500.

To configure the ATM interface on the router, you must install the ATM feature license using the **license install atm** command. To activate or enable the configuration on the IMA interface after the ATM license is installed, use the **license feature atm** command.

For more information about installing licenses, see the Software Activation Configuration Guide, Cisco IOS XE Release 3S.

Note You can create a maximum of 16 IMA groups on each T1/E1 interface module.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**card type** {**t1** | **e1**} *slot* [*bay*] **Example:** `Router(config)# card type e1 0 0`	Specifies the slot and port number of the E1 or T1 interface.
Step 4	**controller** {**t1** | **e1**} *slot/subslot/port* **Example:** `Router(config)# controller e1 0/0/4` **Example:**	Specifies the controller interface on which you want to enable IMA.
Step 5	**clock source internal** **Example:** `Router(config-controller)# clock source internal` **Example:**	Sets the clock source to internal.
Step 6	**ima group** *group-number* **Example:** `Router(config-controller)# ima-group 0 scrambling-payload` **Example:**	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. To add another member link, repeat Step 3 to Step 6 .
Step 7	**exit** **Example:**	Exits the controller interface.

	Command or Action	Purpose
	`Router(config-controller)# exit` **Example:**	
Step 8	**interface ATM***slot*/*subslot*/**IMA** *group-number* **Example:** `Router(config-if)# interface atm0/1/ima0`	Specify the slot location and port of IMA interface group. • *slot*—The location of the ATM IMA interface module. • *group-number*—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, 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. The system toggles the original IMA interface to select a different IMA group ID.
Step 9	**no ip address** **Example:** `Router(config-if)# no ip address`	Disables the IP address configuration for the physical layer interface.
Step 10	**atm bandwidth dynamic** **Example:** `Router(config-if)# atm bandwidth dynamic`	Specifies the ATM bandwidth as dynamic.
Step 11	**no atm ilmi-keepalive** **Example:** `Router(config-if)# no atm ilmi-keepalive`	Disables the Interim Local Management Interface (ILMI) keepalive parameters. ILMI is not supported on the router starting with Cisco IOS XE Release 3.15S.
Step 12	**exit** **Example:**	Exits configuration mode.

	Command or Action	Purpose
	`Router(config)# exit`	

What to do next

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

BGP PIC with TDM Configuration

To configure the TDM pseudowires on the router, see Configuring CEM, on page 7.

To configure BGP PIC on the router, see IP Routing: BGP Configuration Guide, Cisco IOS XE Release 3S (Cisco ASR 900 Series).

See the configuration example, Example: BGP PIC with TDM Configuration, on page 40.

Configuring Structure-Agnostic TDM over Packet (SAToP)

Follow these steps to configure SAToP on the Cisco ASR 903 Series Router:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**controller** [**t1**|**e1**] *slot/sublot* **Example:** `Router(config-controller)# controller t1 0/4`	Configures the T1 or E1 interface.
Step 4	**cem-group** *group-number* {*unframed* | **timeslots** *timeslot*} **Example:** `Router(config-if)# cem-group 4 unframed`	Assigns channels on the T1 or E1 circuit to the CEM channel. This example uses the **unframed** parameter to assign all the T1 timeslots to the CEM channel.
Step 5	**interface cem** *slot/subslot* **Example:**	Defines a CEM group.

	Command or Action	Purpose

	`Router(config)# interface CEM 0/4`
	Example:
	`Router(config-if)# no ip address`
	Example:
	`Router(config-if)# cem 4`	

Step 6	**xconnect** *ip_address* **encapsulation mpls** **Example:** `Router(config-if)# xconnect 10.10.2.204 encapsulation mpls`	Binds an attachment circuit to the CEM interface to create a pseudowire. This example creates a pseudowire by binding the CEM circuit 304 to the remote peer 10.10.2.204.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits configuration mode.

What to do next

Note When creating IP routes for a pseudowire configuration, we recommend that you build a route from the cross-connect address (LDP router-id or loopback address) to the next hop IP address, such as **ip route 10.10.10.2 255.255.255.254 10.2.3.4.**

Configuring Circuit Emulation Service over Packet-Switched Network (CESoPSN)

Procedure

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

	Command or Action	Purpose
Step 3	**controller [e1 | t1]** *slot/subslot* **Example:** `Router(config)# controller e1 0/0` **Example:**	Enters configuration mode for the E1 or T1 controller.
Step 4	**cem-group** *group-number* **timselots** *timeslots* **Example:** `Router(config-controller)# cem-group 5 timeslots 1-24`	Assigns channels on the T1 or E1 circuit to the circuit emulation (CEM) channel. This example uses the **timeslots** parameter to assign specific timeslots to the CEM channel.
Step 5	**exit** **Example:** `Router(config-controller)# exit`	Exits controller configuration.

Step 6	**interface cem** *slot/subslot*
	Example:
	`Router(config)# interface CEM0/5`
	Example:
	`Router(config-if-cem)# cem 5`
	Example:
		Defines a CEM channel.

Step 7	**xconnect** *ip_address* **encapsulation mpls** **Example:** `Router(config-if)# xconnect 10.10.2.204 encapsulation mpls`	Binds an attachment circuit to the CEM interface to create a pseudowire. This example creates a pseudowire by binding the CEM circuit 304 to the remote peer 10.10.2.204.
Step 8	**exit** **Example:** `Router(config-if-cem)# exit`	Exits the CEM interface.
Step 9	**exit** **Example:** `Router(config)# exit`	Exits configuration mode.

Configuring a Clear-Channel ATM Pseudowire

To configure the T1 interface module for clear-channel ATM, follow these steps:

Procedure

	Command or Action	Purpose
Step 1	**controller {t1}** *slot/subslot/port* **Example:** `Router(config)# controller t1 0/4`	Selects the T1 controller for the port you are configuring. **Note** The slot number is always 0.
Step 2	**atm** **Example:** `Router(config-controller)# atm`	Configures the port (interface) for clear-channel ATM. The router creates an ATM interface whose format is atm/*slot* /*subslot* /*port* . **Note** The slot number is always 0.
Step 3	**exit** **Example:** `Router(config-controller)# exit`	Returns you to global configuration mode.
Step 4	**interface atm** *slot/subslot/port* **Example:** `Router(config)#` **`interface atm 0/3/0`**	Selects the ATM interface in Step 2 .
Step 5	**pvc** *vpi/vci* **Example:** `Router(config-if)# pvc 0/40`	Configures a PVC for the interface and assigns the PVC a VPI and VCI. Do not specify 0 for both the VPI and VCI.
Step 6	**xconnect** *peer-router-id vcid* {**encapsulation mpls** | **pseudowire-class** *name* **Example:** `Router(config-if)# xconnect 10.10.2.204 200 encapsulation mpls`	Configures a pseudowire to carry data from the clear-channel ATM interface over the MPLS network.
Step 7	**end** **Example:** `Router(config-if)# end`	Exits configuration mode.

Configuring an ATM over MPLS Pseudowire

ATM over MPLS pseudowires allow you to encapsulate and transport ATM traffic across an MPLS network. This service allows you to deliver ATM services over an existing MPLS network.

The following sections describe how to configure transportation of service using ATM over MPLS:

- Configuring the Controller, on page 23
- Configuring an IMA Interface, on page 24
- Configuring the ATM over MPLS Pseudowire Interface, on page 25

Configuring the Controller

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**card type** {**e1**} *slot/subslot* **Example:** `Router(config)# card type e1 0 0`	Configures IMA on an E1 or T1 interface.
Step 4	**controller {e1}** *slot/subslot* **Example:** `Router(config)# controller e1 0/4`	Specifies the controller interface on which you want to enable IMA.
Step 5	**clock source** {**internal** | **line**} **Example:** `Router(config-controller)# clock source internal`	Sets the clock source to internal.
Step 6	**ima-group** *group-number* **scrambling-payload** **Example:** `Router(config-controller)# ima-group 0 scrambling-payload`	If you want to configure an ATM IMA backhaul, use the **ima-group** command to assign the interface to an IMA group. For a T1 connection, use the **no-scrambling-payload** to disable ATM-IMA cell payload scrambling; for

	Command or Action	Purpose
		an E1 connection, use the **scrambling-payload** parameter to enable ATM-IMA cell payload scrambling. The example assigns the interface to IMA group 0 and enables payload scrambling.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits configuration mode.

Configuring an IMA Interface

If you want to use ATM IMA backhaul, follow these steps to configure the IMA interface.

Note You can create a maximum of 16 IMA groups on each T1/E1 interface module.

Procedure

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

Step 3	**interface ATM** *slot* / **IMA** *group-number*
	Example:
	`Router(config-controller)# interface atm0/ima0`
	Example:
	`Router(config-if)#`
		Specifies the slot location and port of IMA interface group. The syntax is as follows:
		• *slot*—The slot location of the interface module.
		• *group-number*—The group number of the IMA group.
		The example specifies the slot number as 0 and the group number as 0.

	Command or Action	Purpose
		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 4	**no ip address** **Example:** `Router(config-if)# no ip address`	Disables the IP address configuration for the physical layer interface.
Step 5	**atm bandwidth dynamic** **Example:** `Router(config-if)# atm bandwidth dynamic`	Specifies the ATM bandwidth as dynamic.
Step 6	**no atm ilmi-keepalive** **Example:** `Router(config-if)# no atm ilmi-keepalive`	Disables the ILMI keepalive parameters.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits configuration mode.

What to do next

For more information about configuring IMA groups, see the Configuring ATM IMA, on page 16.

Configuring the ATM over MPLS Pseudowire Interface

You can configure ATM over MPLS is several modes according to the needs of your network. Use the appropriate section according to the needs of your network. You can configure the following ATM over MPLS pseudowire types:

- Configuring 1-to-1 VCC Cell Transport Pseudowire, on page 26—Maps a single VCC to a single pseudowire
- Configuring N-to-1 VCC Cell Transport Pseudowire , on page 27—Maps multiple VCCs to a single pseudowire
- Configuring 1-to-1 VPC Cell Transport, on page 27—Maps a single VPC to a single pseudowire
- Configuring ATM AAL5 SDU VCC Transport, on page 29—Maps a single ATM PVC to another ATM PVC
- Configuring a Port Mode Pseudowire, on page 30—Maps one physical port to a single pseudowire connection
- Optional Configurations, on page 31

Note When creating IP routes for a pseudowire configuration, build a route from the xconnect address (LDP router-id or loopback address) to the next hop IP address, such as **ip route 10.10.10.2 255.255.255.255 10.2.3.4.**

Configuring 1-to-1 VCC Cell Transport Pseudowire

A 1-to-1 VCC cell transport pseudowire maps one ATM virtual channel connection (VCC) to a single pseudowire. Complete these steps to configure a 1-to-1 pseudowire.

Note Multiple 1-to-1 VCC pseudowire mapping on an interface is supported.

Mapping a Single PVC to a Pseudowire

To map a single PVC to an ATM over MPLS pseudowire, use the **xconnect** command at the PVC level. This configuration type uses AAL0 and AAL5 encapsulations. Complete these steps to map a single PVC to an ATM over MPLS pseudowire.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface ATM** *slot* / **IMA** *group-number* **Example:** `Router(config-controller)# interface atm0/ima0`	Configures the ATM IMA interface.

	Command or Action	Purpose
Step 4	**pvc** *slot/subslot* **l2transport** **Example:** `Router(config-if-atm)# pvc 0/40 l2transport`	Defines a PVC. Use the **l2transport** keyword to configure the PVC as a layer 2 virtual circuit.
Step 5	**encapsulation aal0** **Example:** `Router(config-if-atm-l2trans-pvc)# encapsulation aal0`	Defines the encapsulation type for the PVC. The default encapsulation type for the PVC is AAL5.
Step 6	**xconnect** *router_ip_address vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.0.0.1 40 encapsulation mpls`	Binds an attachment circuit to the ATM IMA interface to create a pseudowire. This example creates a pseudowire by binding PVC 40 to the remote peer 10.0.0.1.
Step 7	**end** **Example:** `Router(config-if-atm-l2trans-pvp-xconn)# end`	Exits configuration mode.

Configuring N-to-1 VCC Cell Transport Pseudowire

An N-to-1 VCC cell transport pseudowire maps one or more ATM virtual channel connections (VCCs) to a single pseudowire. Complete these steps to configure an N-to-1 pseudowire.

Configuring 1-to-1 VPC Cell Transport

A 1-to-1 VPC cell transport pseudowire maps one or more virtual path connections (VPCs) to a single pseudowire. While the configuration is similar to 1-to-1 VPC cell mode, this transport method uses the 1-to-1 VPC pseudowire protocol and format defined in RFCs 4717 and 4446. Complete these steps to configure a 1-to-1 VPC pseudowire.

Note Multiple 1-to-1 VCC pseudowire mapping on an interface is supported.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**interface ATM** *slot* / **IMA** *group-number*
	Example:
	`Router(config-controller)# interface atm0/ima0`
	Example:
	`Router(config-if)#`
	Example:
		Configures the ATM IMA interface.

Step 4	**atm pvp** *vpi* **l2transport**
	Example:
	`Router(config-if-atm)# atm pvp 10 l2transport`
	Example:
	`Router(config-if-atm-l2trans-pvp)#`
		Maps a PVP to a pseudowire.

Step 5	**xconnect** *peer-router-id vcid* {**encapsulation mpls**
	Example:
	`Router(config-if-atm-l2trans-pvp)# xconnect 10.10.10.2 305 encapsulation mpls`
	Example:
	`Router(config-if-atm-l2trans-pvp-xconn)#`
		Binds an attachment circuit to the ATM IMA interface to create a pseudowire. This example creates a pseudowire by binding the ATM circuit 305 to the remote peer 30.30.30.2.

Step 6	**end** **Example:** `Router(config-if-atm-l2trans-pvp-xconn)# end` **Example:**	Exits the configuration mode.

Configuring ATM AAL5 SDU VCC Transport

An ATM AAL5 SDU VCC transport pseudowire maps a single ATM PVC to another ATM PVC. Follow these steps to configure an ATM AAL5 SDU VCC transport pseudowire.

Procedure

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

Step 3	**interface ATM** *slot* / **IMA** *group-number*
	Example:
	`Router(config-controller)# interface atm0/ima0`
	Example:
	`Router(config-if)#`
	Example:
	Example:
		Configures the ATM IMA interface.

Step 4	**atm pvp** *vpi* **l2transport**
	Example:
	`Router(config-if)# pvc 0/12 l2transport`
	Example:
	`Router(config-if-atm-l2trans-pvc)#`
		Configures a PVC and specifies a VCI or VPI.

Step 5	**encapsulation aal5** **Example:** `Router(config-if-atm-l2trans-pvc)# encapsulation aal5`	Sets the PVC encapsulation type to AAL5. **Note** You must use the AAL5 encapsulation for this transport type.
Step 6	**xconnect** *peer-router-id vcid* **encapsulation mpls**	Binds an attachment circuit to the ATM IMA interface to create a pseudowire. This example

	Command or Action	Purpose
	Example: `Router(config-if-atm-l2trans-pvc)#` **`xconnect 10.10.10.2 125 encapsulation mpls`**	creates a pseudowire by binding the ATM circuit 125 to the remote peer 25.25.25.25.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits configuration mode.

Configuring a Port Mode Pseudowire

A port mode pseudowire allows you to map an entire ATM interface to a single pseudowire connection.

Procedure

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

Step 3	**interface ATM** *slot* / **IMA** *group-number*
	
	Example:
	
	`Router(config-controller)# interface atm0/ima0`
	
	Example:
	
	`Router(config-if)#`
	
	Example:
	
	Example:
		Configures the ATM interface.

Step 4	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-if-atm-l2trans-pvc)#`	Binds an attachment circuit to the ATM IMA interface to create a pseudowire. This example creates a pseudowire by binding the ATM circuit 125 to the remote peer 10.10.10.2.

	Command or Action	Purpose
	`xconnect 10.10.10.2 125 encapsulation mpls`	
Step 5	**exit** **Example:** `Router(config)# exit`	Exits configuration mode.

Optional Configurations

You can apply the following optional configurations to a pseudowire link.

Configuring Cell Packing

Cell packing allows you to improve the efficiency of ATM-to-MPLS conversion by packing multiple ATM cells into a single MPLS packet. Follow these steps to configure cell packing.

Procedure

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

Step 3	**interface ATM** *slot* / **IMA** *group-number*
	Example:
	`Router(config-controller)# interface atm0/ima0`
	Example:
	`Router(config-if)#`
		Configures the ATM interface.

Step 4	**atm mcpt-timers** *timer1 timer2 timer3* **Example:** `Router(config-if)# atm mcpt-timers 1000 2000 3000`	Defines the three Maximum Cell Packing Timeout (MCPT) timers under an ATM interface. The three independent MCPT timers specify a wait time before forwarding a packet.
Step 5	**atm pvp** *vpi* **l2transport** **Example:**	Configures a PVC and specifies a VCI or VPI.

	Command or Action	Purpose

	`Router(config-if)# **pvc 0/12 l2transport**`
	Example:
	`Router(config-if-atm-l2trans-pvc)#`	

Step 6	**encapsulation aal5** **Example:** `Router(config-if-atm-l2trans-pvc)# encapsulation aal5`	Sets the PVC encapsulation type to AAL5. **Note** You must use the AAL5 encapsulation for this transport type.
Step 7	**cell-packing** *maxcells* **mcpt-timer** *timer-number* **Example:** `Router(config-if-atm-l2trans-pvc)# cell-packing 20 mcpt-timer 3`	Specifies the maximum number of cells in PW cell pack and the cell packing timer. This example specifies 20 cells per pack and the third MCPT timer.
Step 8	**end** **Example:** `Router(config-if-atm-l2trans-pvc)# end`	Exits the configuration mode.

Configuring an Ethernet over MPLS Pseudowire

Ethernet over MPLS PWs allow you to transport Ethernet traffic over an existing MPLS network. The router supports EoMPLS pseudowires on EVC interfaces.

For more information about Ethernet over MPLS Pseudowires, see Transportation of Service Using Ethernet over MPLS, on page 6.

Procedure

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

	Command or Action	Purpose
Step 3	**interface** *interface-id* **Example:** `Router(config)# interface gigabitethernet 0/0/4`	Specifies the port on which to create the pseudowire and enters interface configuration mode. Valid interfaces are physical Ethernet ports.
Step 4	**service instance** *number* **ethernet** [*name*] **Example:** `Router(config-if)# service instance 2 ethernet`	Configure an EFP (service instance) and enter service instance configuration) mode. • The *number* is the EFP identifier, an integer from 1 to 4000. • (Optional) **ethernet** *name* is the name of a previously configured EVC. You do not need to use an EVC name in a service instance. **Note** You can use service instance settings such as encapsulation, dot1q, and rewrite to configure tagging properties for a specific traffic flow within a given pseudowire session. For more information, see http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/cether/configuration/xe-3s/asr903/ce-xe-3s-asr903-book/ce-evc.html
Step 5	**encapsulation** {**default** | **dot1q** | **priority-tagged** | **untagged}** **Example:** `Router(config-if-srv)# encapsulation dot1q 2`	Configure encapsulation type for the service instance. • **default**—Configure to match all unmatched packets. • **dot1q**—Configure 802.1Q encapsulation. • **priority-tagged**—Specify priority-tagged frames, VLAN-ID 0 and CoS value of 0 to 7. • **untagged**—Map to untagged VLANs. Only one EFP per port can have untagged encapsulation.
Step 6	**xconnect** *peer-ip-address vc-id* {**encapsulation** {**l2tpv3** [**manual**] | **mpls** [**manual**]} | **pw-class** *pw-class-name* }[**pw-class** *pw-class-name*] [**sequencing** {**transmit** | **receive** | **both**}] **Example:**	Binds the Ethernet port interface to an attachment circuit to create a pseudowire. This example uses virtual circuit (VC) 101 to uniquely identify the PW. Ensure that the remote VLAN is configured with the same VC.

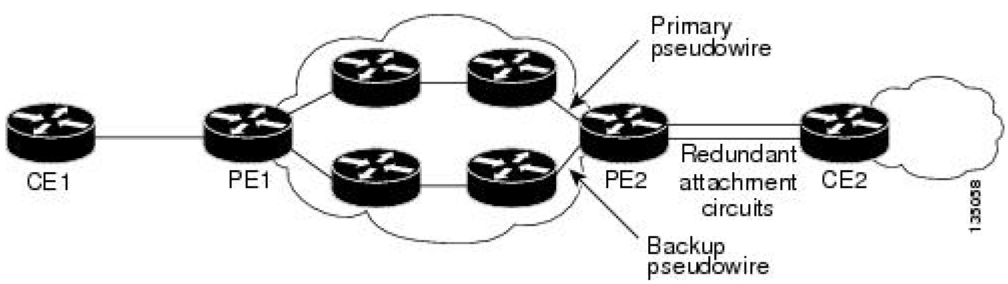
	Command or Action	Purpose
	`Router (config-if-srv)# xconnect 10.1.1.2 101 encapsulation mpls`	**Note** When creating IP routes for a pseudowire configuration, we recommend that you build a route from the xconnect address (LDP router-id or loopback address) to the next hop IP address, such as **ip route 10.10.10.2 255.255.255.255 10.2.3.4**.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits configuration mode.

Configuring Pseudowire Redundancy

A backup peer provides a redundant pseudowire (PW) connection in the case that the primary PW loses connection; if the primary PW goes down, the Cisco ASR 903 Series Router diverts traffic to the backup PW. This feature provides the ability to recover from a failure of either the remote PE router or the link between the PE router and CE router.

Figure 3: Pseudowire Redundancy, on page 34 shows an example of pseudowire redundancy.

Figure 3: Pseudowire Redundancy



Note You must configure the backup pseudowire to connect to a router that is different from the primary pseudowire.

Follow these steps to configure a backup peer:

Procedure

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

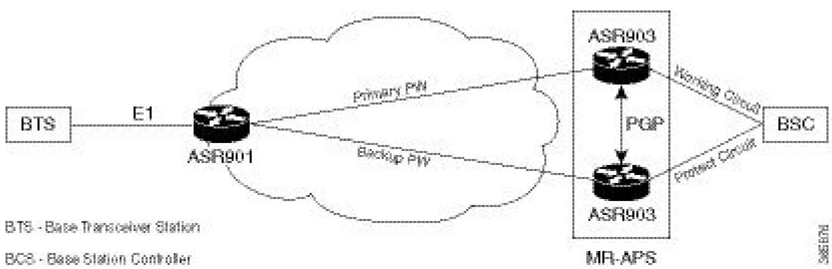
	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config)# pseudowire-class mpls`	Specify the name of a Layer 2 pseudowire class and enter pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies MPLS encapsulation.
Step 5	**interface serial** *slot*/*subslot*/*port* **Example:** `Router(config)# interface serial0/0`	Enters configuration mode for the serial interface. **Note** The slot number is always 0.
Step 6	**backup delay** *enable-delay* {*disable-delay* | **never**} **Example:** `Router(config)# backup delay 0 10`	Configures the backup delay parameters. Where: • *enable-delay*—Time before the backup PW takes over for the primary PW. • *disable-delay*—Time before the restored primary PW takes over for the backup PW. • **never**—Disables switching from the backup PW to the primary PW.
Step 7	**xconnect** *router-id* **encapsulation mpls** **Example:** `Router(config-if)# xconnect 10.10.10.2 101 encapsulation mpls`	Binds the Ethernet port interface to an attachment circuit to create a pseudowire.
Step 8	**backup peer** *peer-router-ip-address vcid* [**pw-class** *pw-class name*] **Example:** `Router(config)# backup peer 10.10.10.1 104 pw-class pw1`	Defines the address and VC of the backup peer.
Step 9	**exit** **Example:** `Router(config)# exit`	Exits configuration mode.

Pseudowire Redundancy with Uni-directional Active-Active

Pseudowire redundancy with uni-directional active-active feature configuration allows, pseudowires (PW) on both the working and protect circuits to remain in UP state to allow traffic to flow from the upstream. The **aps l2vpn-state detach** command and **redundancy all-active replicate** command is introduced to configure uni-directional active-active pseudowire redundancy.

In pseudowire redundancy Active-Standby mode, the designation of the active and standby pseudowires is decided either by the endpoint PE routers or by the remote PE routers when configured with MR-APS. The active and standby routers communicate via Protect Group Protocol (PGP) and synchronize their states. The PEs are connected to a Base Station Controller (BSC). APS state of the router is communicated to the Layer2 VPN, and is therby coupled with the pseudowire status .

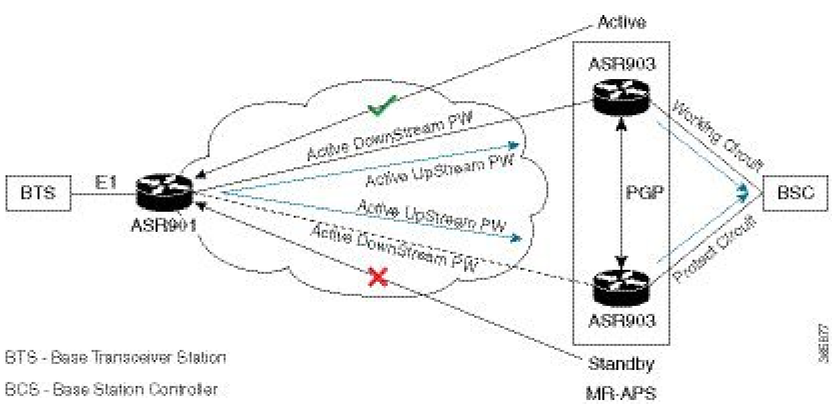
Figure 4: Pseudowire Redundancy with MR-APS



BSC monitors the status of the incoming signal from the working and protect routers. In the event of a swithover at the BSC, the BSC fails to inform the PE routers, hence causing traffic drops.

With pseudowire redundancy Active-Active configuration, the traffic from the upstream is replicated and transmitted over both the primary and backup pseudowires. PE routers forwards the received traffic to the working and protect circuits. The BSC receives the same traffic on both the circuits and selects the better Rx link, ensuring the traffic is not dropped.

Figure 5: Pseudowire Redundancy with Uni-directional Active-Active



Note If teh ASR 900 router is configured with the **aps l2vpn-state detach** command but, the ASR 901 router is not enabled with **redundancy all-active replicate** command, the protect PW is active after APS switchover. On the ASR 901 router, the PW state is UP and the data path status displays standby towards protect node. On an APS switchover on the ASR 900 router, the status is not communicated to ASR 901 router, and the VC data path state towards the protect node remains in the standby state.

Restrictions

The following restrictions apply on the router:

- If the **aps l2vpn-state detach** command is enabled on the ASR 900 router, but the **redundancy all-active replicate** command *not* enabled on the ASR 901 router, the pseudowire status on the router displays UP, and the data path status for the protect node state displays Standby.
- After APS switchover on the ASR 900 router, the status is *not* communicated to ASR 901 router, and the virtual circuit data path state towards the protect node remains in the Standby state.
- The **aps l2vpn-state detach** command takes effect after a controller **shutdown** command, followed by a **no shutown** command is performed. Alternately, the command can be configured when the controller is in shut state.
- The **status peer topology dual-homed** command in pseudowire-class configuration mode should *not* be configured on the ASR 900 router, irrespective of unidirectional or bidirectional mode. The command *must* be configured on the ASR 901 router.
- Traffic outages from the BSC to the BTS on PGP and ICRM failures at the working Active node, is same as the configured hold time.

Note APS switchover may be observed on the protect node, when PGP failure occurs on the working Active node.

- Convergence may be observed on performing a power cycle on the Active (whether on the protect or working) node. The observed convergence is same as the configured hold time.

Configuring Pseudowire Redundancy Active-Active— Protocol Based

```
encapsulation mpls
status peer topology dual-homed

controller E1 0/1
framing unframed
cem-group 8 unframed
```

Configuring the Working Controller for MR-APS with Pseudowire Redundancy Active-Active

The following configuration shows pseudowire redundancy active-active for MR-APS working controller:

```
controller sonet 0/1/0
aps group 2
aps adm
aps working 1
aps timers 1 3
aps l2vpn-state detach
aps hspw-icrm-grp 1
```

Configuring the Protect Controller for MR-APS with Pseudowire Redundancy Active-Active

Following example shows pseudowire redundancy active-active on MR-APS protect controller:

```
controller sonet 0/1/0
aps group 2
aps adm
aps unidirectional
aps protect 10 10.10.10.1
aps timers 1 3
aps l2vpn-state detach
aps hspw-icrm-grp 1
```

Verifying the Interface Configuration

You can use the following commands to verify your pseudowire configuration:

- **show cem circuit**—Displays information about the circuit state, administrative state, the CEM ID of the circuit, and the interface on which it is configured. If **xconnect** is configured under the circuit, the command output also includes information about the attached circuit.

```
Router# show cem circuit
 ?

  <0-504>    CEM ID
  detail     Detailed information of cem ckt(s)
  interface  CEM Interface
  summary    Display summary of CEM ckts
  |          Output modifiers
Router# show cem circuit

CEM Int.       ID   Line      Admin     Circuit         AC
--------------------------------------------------------------
CEM0/1/0       1    UP        UP        ACTIVE          --/--
CEM0/1/0       2    UP        UP        ACTIVE          --/--
CEM0/1/0       3    UP        UP        ACTIVE          --/--
```

```
CEM0/1/0        4     UP         UP         ACTIVE          --/--
CEM0/1/0        5     UP         UP         ACTIVE          --/--
```

- **show cem circuit**—Displays the detailed information about that particular circuit.

```
Router# show cem circuit 1

CEM0/1/0, ID: 1, Line State: UP, Admin State: UP, Ckt State: ACTIVE
Idle Pattern: 0xFF, Idle cas: 0x8, Dummy Pattern: 0xFF
Dejitter: 5, Payload Size: 40
Framing: Framed, (DS0 channels: 1-5)
Channel speed: 56
CEM Defects Set
Excessive Pkt Loss RatePacket Loss
Signalling: No CAS
Ingress Pkts:    25929                 Dropped:             0
Egress Pkts:     0                     Dropped:             0
CEM Counter Details
Input Errors:    0                     Output Errors:       0
Pkts Missing:    25927                 Pkts Reordered:      0
Misorder Drops:  0                     JitterBuf Underrun:  1
Error Sec:       26                    Severly Errored Sec: 26
Unavailable Sec: 5                     Failure Counts:      1
Pkts Malformed:  0
```

- **show cem circuit summary**—Displays the number of circuits which are up or down per interface basis.

```
Router# show cem circuit summary

CEM Int.       Total Active  Inactive
--------------------------------------
CEM0/1/0       5      5       0
```

- **show running configuration**—The **show running configuration** command shows detail on each CEM group.

Configuration Examples

The following sections contain sample pseudowire configurations.

Example: CEM Configuration

The following example shows how to add a T1 interface to a CEM group as a part of a SAToP pseudowire configuration. For more information about how to configure pseudowires, see Configuring Pseudowire, on page 1

Note This section displays a partial configuration intended to demonstrate a specific feature.

```
controller T1 0/0/0
 framing unframed
 clock source internal
 linecode b8zs
 cablelength short 110
 cem-group 0 unframed
```

```
interface CEM0/0/0
 no ip address
 cem 0
  xconnect 18.1.1.1 1000 encapsulation mpls
```

Example: BGP PIC with TDM Configuration

CEM Configuration

```
pseudowire-class pseudowire1
encapsulation mpls
control-word
no status control-plane route-watch
!
controller SONET 0/2/3
description connected to CE2 SONET 4/0/0
framing sdh
clock source line
aug mapping au-4
!
au-4 1 tug-3 1
  mode c-12
  tug-2 1 e1 1 cem-group 1101 unframed
  tug-2 1 e1 1 framing unframed
  tug-2 1 e1 2 cem-group 1201 timeslots 1-10
 !
au-4 1 tug-3 2
  mode c-12
  tug-2 5 e1 1 cem-group 1119 unframed
  tug-2 5 e1 1 framing unframed
  tug-2 5 e1 2 cem-group 1244 timeslots 11-20
 !
au-4 1 tug-3 3
  mode c-12
  tug-2 5 e1 3 cem-group 1130 unframed
  tug-2 5 e1 3 framing unframed
  tug-2 7 e1 3 cem-group 1290 timeslots 21-30
!
interface CEM0/2/3
no ip address
cem 1101
  xconnect 17.1.1.1 1101 encapsulation mpls pw-class pseudowire1
 !
cem 1201
  xconnect 17.1.1.1 1201 encapsulation mpls pw-class pseudowire1
 !
cem 1119
  xconnect 17.1.1.1 1119 encapsulation mpls pw-class pseudowire1
 !
cem 1244
  xconnect 17.1.1.1 1244 encapsulation mpls pw-class pseudowire1
 !
cem 1130
  xconnect 17.1.1.1 1130 encapsulation mpls pw-class pseudowire1
 !
cem 1290
  xconnect 17.1.1.1 1290 encapsulation mpls pw-class pseudowire1
```

BGP PIC Configuration

```
cef table output-chain build favor convergence-speed
!
router bgp 1
bgp log-neighbor-changes
bgp graceful-restart
neighbor 18.2.2.2 remote-as 1
neighbor 18.2.2.2 update-source Loopback0
neighbor 18.3.3.3 remote-as 1
neighbor 18.3.3.3 update-source Loopback0
!
address-family ipv4
  bgp additional-paths receive
  bgp additional-paths install
  bgp nexthop trigger delay 0
  network 17.5.5.5 mask 255.255.255.255
  neighbor 18.2.2.2 activate
  neighbor 18.2.2.2 send-community both
  neighbor 18.2.2.2 send-label
  neighbor 18.3.3.3 activate
  neighbor 18.3.3.3 send-community both
  neighbor 18.3.3.3 send-label
exit-address-family
```

Example: BGP PIC with TDM-PW Configuration

This section lists the configuration examples for BGP PIC with TDM and TDM–Pseudowire.

The below configuration example is for BGP PIC with TDM:

```
router bgp 1
neighbor 18.2.2.2 remote-as 1
neighbor 18.2.2.2 update-source Loopback0
neighbor 18.3.3.3 remote-as 1
neighbor 18.3.3.3 update-source Loopback0
!
address-family ipv4
  bgp additional-paths receive
  bgp additional-paths install
  bgp nexthop trigger delay 6
  neighbor 18.2.2.2 activate
  neighbor 18.2.2.2 send-community both
  neighbor 18.2.2.2 send-label
  neighbor 18.3.3.3 activate
  neighbor 18.3.3.3 send-community both
  neighbor 18.3.3.3 send-label
  neighbor 26.1.1.2 activate
exit-address-family
!
address-family vpnv4
  bgp nexthop trigger delay 7
  neighbor 18.2.2.2 activate
  neighbor 18.2.2.2 send-community extended
  neighbor 18.3.3.3 activate
  neighbor 18.3.3.3 send-community extended
exit-address-family
```

The below configuration example is for BGP PIC with TDM PW:

```
pseudowire-class pseudowire1
```

```
encapsulation mpls
control-word
no status control-plane route-watch
status peer topology dual-homed
!
Interface CEM0/0/0
cem 1
   xconnect 17.1.1.1 4101 encapsulation mpls pw-class pseudowire1
```

Example: ATM IMA Configuration

The following example shows how to add a T1/E1 interface to an ATM IMA group as a part of an ATM over MPLS pseudowire configuration. For more information about how to configure pseudowires, see Configuring Pseudowire, on page 1

Note This section displays a partial configuration intended to demonstrate a specific feature.

```
controller t1 4/0/0
 ima-group 0
 clock source line
interface atm4/0/ima0
 pvc 1/33 l2transport
  encapsulation aal0
  xconnect 10.0.0.1 33 encapsulation mpls
```

Example: ATM over MPLS

The following sections contain sample ATM over MPLS configurations:

Cell Packing Configuration Examples

The following sections contain sample ATM over MPLS configuration using Cell Relay:

VC Mode

CE 1 Configuration

```
interface Gig4/3/0
no negotiation auto
load-interval 30
interface Gig4/3/0
ip address 20.1.1.1 255.255.255.0
interface ATM4/2/4
no shut
exit
!
interface ATM4/2/4.10 point
ip address 50.1.1.1 255.255.255.0
pvc 20/101
encapsulation aal5snap
!
ip route 30.1.1.2 255.255.255.255 50.1.1.2
```

CE 2 Configuration

```
interface Gig8/8
no negotiation auto
load-interval 30
interface Gig8/8
ip address 30.1.1.1 255.255.255.0
interface ATM6/2/1
no shut
!
interface ATM6/2/1.10 point
ip address 50.1.1.2 255.255.255.0
pvc 20/101
encapsulation aal5snap
!
ip route 20.1.1.2 255.255.255.255 50.1.1.1
```

PE 1 Configuration

```
interface Loopback0
ip address 192.168.37.3 255.255.255.255
!
interface ATM0/0/0
no shut
!
interface ATM0/0/0
atm mcpt-timers 150 1000 4095
interface ATM0/0/0.10 point
pvc 20/101 l2transport
encapsulation aal0
cell-packing 20 mcpt-timer 1
xconnect 192.168.37.2 100 encapsulation mpls
!
interface Gig0/3/0
no shut
ip address 40.1.1.1 255.255.0.0
mpls ip
!
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
mpls ldp graceful-restart
router ospf 1
network 40.1.0.0 0.0.255.255 area 1
network 192.168.37.0 0.0.0.255 area 1
nsf
```

PE 2 Configuration

```
interface Loopback0
ip address 192.168.37.2 255.255.255.255
!
interface ATM9/3/1
no shut
!
interface ATM9/3/1
atm mcpt-timers 150 1000 4095
interface ATM9/3/1.10 point
pvc 20/101 l2transport
encapsulation aal0
cell-packing 20 mcpt-timer 1
```

```
xconnect 192.168.37.3 100 encapsulation mpls
!
interface Gig6/2
no shut
ip address 40.1.1.2 255.255.0.0
mpls ip
!
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
mpls ldp graceful-restart
router ospf 1
network 40.1.0.0 0.0.255.255 area 1
network 192.168.37.0 0.0.0.255 area 1
nsf
```

VP Mode

CE 1 Configuration

```
interface Gig4/3/0
no negotiation auto
load-interval 30
interface Gig4/3/0
ip address 20.1.1.1 255.255.255.0
interface ATM4/2/4
!
interface ATM4/2/4.10 point
ip address 50.1.1.1 255.255.255.0
pvc 20/101
encapsulation aal5snap
!
ip route 30.1.1.2 255.255.255.255 50.1.1.2
```

CE 2 Configuration

```
!
interface Gig8/8
no negotiation auto
load-interval 30
interface Gig8/8
ip address 30.1.1.1 255.255.255.0
interface ATM6/2/1
no shut
!
interface ATM6/2/1.10 point
ip address 50.1.1.2 255.255.255.0
pvc 20/101
encapsulation aal5snap
!
ip route 20.1.1.2 255.255.255.255 50.1.1.1
```

PE 1 Configuration

```
interface Loopback0
ip address 192.168.37.3 255.255.255.255
!
interface ATM0/0/0
no shut
```

```
!
interface ATM0/0/0
atm mcpt-timers 150 1000 4095
interface ATM0/0/0.50 multipoint
atm pvp 20 l2transport
cell-packing 10 mcpt-timer 1
xconnect 192.168.37.2 100 encapsulation mpls
!
interface Gig0/3/0
no shut
ip address 40.1.1.1 255.255.0.0
mpls ip
!
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
mpls ldp graceful-restart
router ospf 1
network 40.1.0.0 0.0.255.255 area 1
network 192.168.37.0 0.0.0.255 area 1
nsf
```

PE 2 Configuration

```
!
interface Loopback0
ip address 192.168.37.2 255.255.255.255
!
interface ATM9/3/1
no shut
!
interface ATM9/3/1
atm mcpt-timers 150 1000 4095
interface ATM9/3/1.50 multipoint
atm pvp 20 l2transport
cell-packing 10 mcpt-timer 1
xconnect 192.168.37.3 100 encapsulation mpls
!
interface Gig6/2
no shut
ip address 40.1.1.2 255.255.0.0
mpls ip
!
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
mpls ldp graceful-restart
router ospf 1
network 40.1.0.0 0.0.255.255 area 1
network 192.168.37.0 0.0.0.255 area 1
nsf
```

Cell Relay Configuration Examples

The following sections contain sample ATM over MPLS configuration using Cell Relay:

VC Mode

CE 1 Configuration

```
!
interface gigabitethernet4/3/0
no negotiation auto
load-interval 30
interface gigabitethernet4/3/0
ip address 20.1.1.1 255.255.255.0
!
interface ATM4/2/4
!
interface ATM4/2/4.10 point
ip address 50.1.1.1 255.255.255.0
pvc 20/101
encapsulation aal5snap
!
ip route 30.1.1.2 255.255.255.255 50.1.1.2
!
```

CE 2 Configuration

```
interface gigabitethernet8/8
no negotiation auto
load-interval 30
interface gigabitethernet8/8
ip address 30.1.1.1 255.255.255.0
interface ATM6/2/1
!
interface ATM6/2/1.10 point
ip address 50.1.1.2 255.255.255.0
pvc 20/101
encapsulation aal5snap
!
ip route 20.1.1.2 255.255.255.255 50.1.1.1
```

PE 1 Configuration

```
!
interface Loopback0
ip address 192.168.37.3 255.255.255.255
!
interface ATM0/0/0
!
interface ATM0/0/0.10 point
pvc 20/101 l2transport
encapsulation aal0
xconnect 192.168.37.2 100 encapsulation mpls
!
interface gigabitethernet0/3/0
ip address 40.1.1.1 255.255.0.0
mpls ip
!
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
mpls ldp graceful-restart
router ospf 1
network 40.1.0.0 0.0.255.255 area 1
```

```
network 192.168.37.0 0.0.0.255 area 1
nsf
```

PE 2 Configuration

```
!
interface Loopback0
ip address 192.168.37.2 255.255.255.255
!
interface ATM9/3/1
!
interface ATM9/3/1.10 point
pvc 20/101 l2transport
encapsulation aal0
xconnect 192.168.37.3 100 encapsulation mpls
!
interface gigabitethernet6/2
ip address 40.1.1.2 255.255.0.0
mpls ip
!
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
mpls ldp graceful-restart
router ospf 1
network 40.1.0.0 0.0.255.255 area 1
network 192.168.37.0 0.0.0.255 area 1
nsf
```

VP Mode

CE 1 Configuration

```
!
interface gigabitethernet4/3/0
no negotiation auto
load-interval 30
interface gigabitethernet4/3/0
ip address 20.1.1.1 255.255.255.0
!
interface ATM4/2/4
!
interface ATM4/2/4.10 point
ip address 50.1.1.1 255.255.255.0
pvc 20/101
encapsulation aal5snap
!
ip route 30.1.1.2 255.255.255.255 50.1.1.2
```

CE 2 Configuration

```
!
interface gigabitethernet8/8
no negotiation auto
load-interval 30
interface gigabitethernet8/8
ip address 30.1.1.1 255.255.255.0
interface ATM6/2/1
!
```

```
interface ATM6/2/1.10 point
ip address 50.1.1.2 255.255.255.0
pvc 20/101
encapsulation aal5snap
!
ip route 20.1.1.2 255.255.255.255 50.1.1.1
```

PE 1 Configuration

```
interface Loopback0
ip address 192.168.37.3 255.255.255.255
!
!
interface ATM0/0/0
interface ATM0/0/0.50 multipoint
atm pvp 20 l2transport
xconnect 192.168.37.2 100 encapsulation mpls
!
interface gigabitethernet0/3/0
ip address 40.1.1.1 255.255.0.0
mpls ip
!
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
mpls ldp graceful-restart
router ospf 1
network 40.1.0.0 0.0.255.255 area 1
network 192.168.37.0 0.0.0.255 area 1
nsf
```

PE 2 Configuration

```
interface Loopback0
ip address 192.168.37.2 255.255.255.255
!
!
interface ATM9/3/1
interface ATM9/3/1.50 multipoint
atm pvp 20 l2transport
xconnect 192.168.37.3 100 encapsulation mpls
!
interface gigabitethernet6/2
ip address 40.1.1.2 255.255.0.0
mpls ip
!
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
mpls ldp graceful-restart
router ospf 1
network 40.1.0.0 0.0.255.255 area 1
network 192.168.37.0 0.0.0.255 area 1
nsf
```

Example: Ethernet over MPLS

PE 1 Configuration

```
!
mpls label range 16 12000 static 12001 16000
mpls label protocol ldp
mpls ldp neighbor 10.1.1.1 targeted ldp
mpls ldp graceful-restart
multilink bundle-name authenticated
!
!
!
!
redundancy
 mode sso
!
!
!
ip tftp source-interface GigabitEthernet0
!
!
interface Loopback0
 ip address 10.5.5.5 255.255.255.255
!
interface GigabitEthernet0/0/4
 no ip address
 negotiation auto
!
 service instance 2 ethernet
  encapsulation dot1q 2
  xconnect 10.1.1.1 1001 encapsulation mpls
 !
 service instance 3 ethernet
  encapsulation dot1q 3
  xconnect 10.1.1.1 1002 encapsulation mpls
 !
!
interface GigabitEthernet0/0/5
 ip address 172.7.7.77 255.0.0.0
 negotiation auto
 mpls ip
 mpls label protocol ldp
!
router ospf 1
 router-id 5.5.5.5
 network 5.5.5.5 0.0.0.0 area 0
 network 172.0.0.0 0.255.255.255 area 0
 network 10.33.33.33 0.0.0.0 area 0
 network 192.0.0.0 0.255.255.255 area 0
!
```

PE 2 Configuration

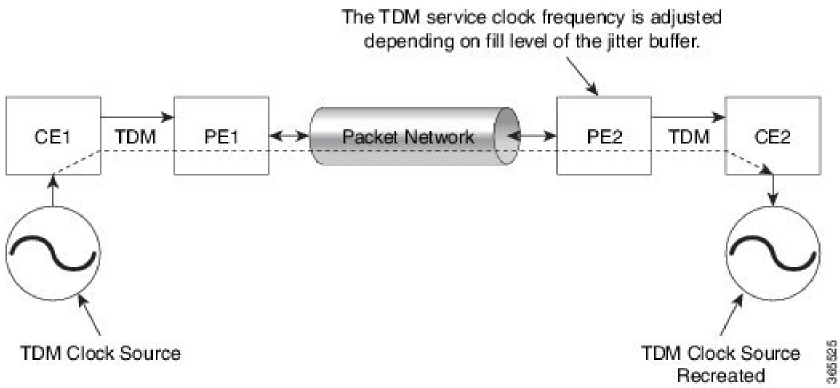
```
!
mpls label range 16 12000 static 12001 16000
mpls label protocol ldp
mpls ldp neighbor 10.5.5.5 targeted ldp
mpls ldp graceful-restart
multilink bundle-name authenticated
```

```
!
!
redundancy
 mode sso
!
!
!
ip tftp source-interface GigabitEthernet0
!
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
interface GigabitEthernet0/0/4
 no ip address
 negotiation auto
!
 service instance 2 ethernet
  encapsulation dot1q 2
  xconnect 10.5.5.5 1001 encapsulation mpls
 !
 service instance 3 ethernet
  encapsulation dot1q 3
  xconnect 10.5.5.5 1002 encapsulation mpls
 !
!
interface GigabitEthernet0/0/5
 ip address 172.7.7.7 255.0.0.0
 negotiation auto
 mpls ip
 mpls label protocol ldp
!
router ospf 1
 router-id 10.1.1.1
 network 10.1.1.1 0.0.0.0 area 0
 network 172.0.0.0 0.255.255.255 area 0
 network 10.33.33.33 0.0.0.0 area 0
 network 192.0.0.0 0.255.255.255 area 0
!
```

Adaptive Clock Recovery (ACR)

Adaptive Clock Recovery (ACR) is an averaging process that negates the effect of random packet delay variation and captures the average rate of transmission of the original bit stream. ACR recovers the original clock for a synchronous data stream from the actual payload of the data stream. In other words, a synchronous clock is derived from an asynchronous packet stream. ACR is a technique where the clock from the TDM domain is mapped through the packet domain, but is most commonly used for Circuit Emulation (CEM). ACR is supported on unframed and framed modes of SAToP.

Note Framing type should be maintained same in all routers end to end.



Benefits of ACR for 8 T1/E1 Interface Module

- Customer-edge devices (CEs) can have different clocks from that of the Provide-edge devices (PEs). Every T1/E1 interface module supports eight pseudowires (or the derived clocks).

Prerequisites for ACR Configuration in 8 T1/E1 Interface Module

- Ensure that CEM is configured before configuring the adaptive clock recovery.
- The following must be configured before configuring the ACR:
  - The remote Customer Equipment and the remote Provider Edge device. These can be configured by using the clock source internal and the clock source line commands under the T1/E1 controller.
  - The controller on the local Customer Equipment connected to the ACR router by using the **clock source line**command.
  - PRC or PRS reference clock from a GPS reference to the remote Customer Equipment or remote CEM Provider Edge device.

Restrictions for ACR on 8 T1/E1 Interface Module

- ACR is supported only on the 8-port T1/E1 interface module (A900-IMA8D). It is not supported on the 16-port T1/E1 interface module (A900-IMA16D), the 32-port T1/E1 interface module (A900-IMA32D), or the 4-port OC3 interface module (A900-IMA4OS).
- ACR is supported only for unframed and framed CEM (SAToP) and for fully-framed CEM (CESoPSN). Fully-framed refers to all the timeslots of T1 (1-24) or E1 (1-31) interfaces.
- ACR is supported only for CEM circuits with MPLS PW encapsulation. ACR is not supported for CEM circuits with UDP or IP PW encapsulation.
- The clock recovered by an ACR clock for a CEM circuit is local to that CEM circuit. The recovered clock cannot be introduced to another circuit and also cannot be introduced to the system clock as a frequency input source.
- The clock ID should be unique for the entire device.
- When a CEM group is configured, dynamic change in clock source is not allowed.

- Physical or soft IM OIR causes the APS switchover time to be higher (500 to 600 ms). Shut or no shut of the port and removal of the active working or protect also cause the APS switchover time to be high.

  To overcome these issues, force the APS switchover.

Configuring ACR for T1 Interfaces for SAToP

To configure the clock on T1/E1 interfaces for SAToP in controller mode:

```
enable
configure terminal
controller t1 0/4/3
clock source recovered 15
cem-group 20 unframed
exit
```

To configure the clock recovery on T1/E1 interfaces in global configuration mode:

```
recovered-clock 0 4
clock recovered 15 adaptive cem 3 20
exit
```

Note The clock source recovered configuration on the controller must be completed before configuring the clock recovery in global configuration mode.

Note On the controller, the clock source should be configured before CEM group is configured.

Note Follow a similar procedure to configure to configure CEM ACR for E1 Interfaces for SAToP. Also, follow a similar procedure to configure CEM ACR for T1 and E1 Interfaces for CESoPSN. Use **cem-group** *circuit-id* **timeslots** *<1-24>* | *<1-31>* command instead of **cem-group** *circuit-id unframed* command for the configuration depending on T1 or E1 controller.

To remove the clock configuration in ACR, you must remove the recovery clock configuration in global configuration mode, then remove the CEM circuit, and finally remove the clock source recovered configuration under the controller.

Note For the 8-port T1/E1 interface module (A900-IMA8D), the configuration or unconfiguration of the clock source recovered is not supported when the cem-group is already configured on the controller. To modify the clock source, you should remove the CEM group configuration from the controller.

Verifying the ACR Configuration of T1 Interfaces for SAToP

Important Notes

- When multiple ACR clocks are provisioned and if the core network or PSN traffic load primarily has fixed packet rate and fixed size packets, the states of one or more ACR clocks might flap between Acquiring and Acquired states and might not be stable in Acquired state.

  This happens because of the "beating" phenomenon and is documented in *ITU-T G.8261 - Timing and synchronization aspects in packet networks*.

  This is an expected behavior.

- After an ACR clock is provisioned and starts recovering the clock, a waiting period of 15-20 minutes is mandatory before measuring MTIE for the recovered clock.

  This behavior is documented in *ITU-T G.8261 Timing and synchronization aspects in packet networks Appendix 2*.

- When the input stream of CEM packets from the core network or PSN traffic is lost or has many errors, the ACR clock enters the HOLDOVER state. In this state, the ACR clock fails to provide an output clock on the E1/T1 controller. Hence, during the HOLDOVER state, MTIE measurement fails.

  This is an expected behavior.

- When the clock output from the clock master or GPS reference flaps or fails, the difference in the characteristics between the holdover clock at the source device and the original GPS clock may result in the ACR algorithm failing to recover clock for a transient period. The MTIE measurement for the ACR clock fails during this time. After this transient period, a fresh MTIE measurement is performed. Similarly, when the GPS clock recovers, for the same difference in characteristics, ACR fails to recover clock and MTIE fails for a transient period.

  This is an expected behavior.

- When large-sized packets are received along with the CEM packets by the devices in the core network or PSN traffic, CEM packets may incur delay with variance in delay. As ACR is susceptible to delay and variance in delay, MTIE measurement may fail. This behavior is documented in *ITU-T G.8261 section 10*.

  This is an expected behavior.

- For a provisioned ACR clock that is in Acquired state, if the ACR clock configuration under the recovered-clock global configuration mode is removed and then reconfigured, the status of the ACR clock may initially be ACQUIRED and not FREERUN and then move to Acquiring. This happens because the ACR clock is not fully unprovisioned until the CEM circuit and the controller clock source recovered configuration are removed. Hence, the clock starts from the old state and then re-attempts to recover the clock.

  This is an expected behavior.

Use the **show recovered-clock** command to verify the ACR of T1 interfaces for SAToP:

```
Router#show recovered-clock
Recovered clock status for subslot 0/1
----------------------------------------
Clock Type Mode Port CEM Status Frequency Offset(ppb)
1 T1/E1 ADAPTIVE 3 1 ACQUIRED 100
```

Use the **show running-config** command to verify the recovery of adaptive clock of T1 interfaces:

```
Router#show running-config
controller T1 0/1/2
clock source recovered 1
cem-group 1 unframed
```

```
interface CEM0/1/3
cem 1
no ip address
xconnect 2.2.2.2 10
encapsulation mpls
```

Associated Commands

Commands	Links
cem-group	http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/interface/command/ir-cr-book/ir-c1.html#wp2440628600
clock source	http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/interface/command/ir-cr-book/ir-c2.html#wp3848511150
clock recovered adaptive cem	http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/interface/command/ir-cr-book/ir-c2.html#wp8894393830
controller t1	http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/interface/command/ir-cr-book/ir-c2.html#wp1472647421
recovered-clock	http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/interface/command/ir-cr-book/ir-c2.html

CHAPTER 2

Automatic Protection Switching Configuration

> **Note** Automatic Protection Switching is *not* supported on the Cisco ASR 900 RSP3 module.

Automatic protection switching (APS) is a protection mechanism for SONET networks that enables SONET connections to switch to another SONET circuit when a circuit failure occurs. A protect interface serves as the backup interface for the working interface. When the working interface fails, the protect interface quickly assumes its traffic load.

Automatic Protection Switching

The protection mechanism used for this feature is "1+1, Bidirectional, nonrevertive" as described in the Bellcore publication "TR-TSY-000253, SONET Transport Systems; Common Generic Criteria, Section 5.3." In the 1+1 architecture, there is one working interface (circuit) and one protect interface, and the same payload from the transmitting end is sent to both the receiving ends. The receiving end decides which interface to use. The line overhead (LOH) bytes (K1 and K2) in the SONET frame indicate both status and action.

The protect interface is configured with the IP address of the router that has the working interface. The APS Protect Group Protocol, which runs on top of UDP, provides communication between the process controlling the working interface and the process controlling the protect interface. Using this protocol, interfaces can be switched because of a router failure, degradation or loss of channel signal, or manual intervention. In bidirectional mode, the receive and transmit channels are switched as a pair.

Two SONET/SDH connections are required to support APS. In a telco environment, the SONET/SDH circuits must be provisioned as APS. You must also provision the operation (for example, 1+1), mode (for example, bidirectional), and revert options (for example, no revert). If the SONET/SDH connections are homed on two separate routers (the normal configuration), an out of band (OOB) communications channel between the two routers needs to be set up for APS communication.

When configuring APS, we recommend that you configure the working interface first. Normal operation with 1+1 operation is to configure it as a working interface. Also configure the IP address of the interface being used as the APS OOB communications path.

APS uses Protect Group Protocol (PGP) between working and protect interfaces. The protect interface APS configuration should include an IP address of a loopback interface on the same router to communicate with the working interface using PGP. Using the PGP, POS interfaces can be switched in case of a degradation or loss of channel signal, or manual intervention. In bidirectional mode, the receive and transmit channels are switched as a pair.

In bidirectional APS the local and the remote connections negotiate the ingress interface to be selected for the data path. The egress interface traffic is not transmitted to both working and protect interfaces.

Inter Chassis Redundancy Manager

ICRM provides these capabilities for stateful MLPPP with MR-APS Inter-Chassis Redundancy implementation:

- Node health monitoring for complete node, PE, or box failure detection. ICRM also communicates failures to the applications registered with an ICRM group.
- Reliable data channels to transfer the state information.
- Detects active RP failure as node failure and notifies the controllers.

ICRM on the standby RP re-establishes the communication channel with peer node if the active RP fails.

For instructions on how to configure ICRM, see Stateful MLPPP Configuration with MR-APS Inter-Chassis Redundancy.

Limitations

- Starting Cisco IOS XE Release 3.11, APS is supported with CES.
- The APS group number range supported on the RSP2 module in aps group *group-number* **acr** command is 1-191.
- APS is *not* supported with ATM.
- APS is *not* supported with IMA.
- APS is *not* supported with POS.
- APS supports HDLC, PPP, and MLPPP encapsulation.
- ATM Layer 2 AAL0 and AAL5 encapsulation types are supported.
- APS is only supported on MLP and serial interfaces on the OC-3 interface module.
- You can perform a force switchover to overcome the following limitations:
  - An interface module OIR, either through physical or software causes the APS switchover time to be higher (500-600ms).

- Shut or no shut of the port may lead to higher APS switchover time.
- Removal of the active working or protect interface may lead to higher APS switchover time.

Automatic Protection Switching Interfaces Configuration

The following sections describe how to configure APS interfaces:

Note We recommend that you configure the working interface before the protected interface in order to prevent the protected interface from becoming the active interface and disabling the working interface.

Note For information about configuring optical interfaces for the first time, see the Cisco ASR 903 Series Router Chassis Configuration Guide.

Configuring a Working Interface

To configure a working interface, use the following commands beginning in global configuration mode.

Before you begin

To configure the controller in SDH mode, see Configuring Optical Interface Modules.

Procedure

	Command or Action	Purpose
Step 1	**controller sonet** *slot* / *port-adapter* / *port* **Example:** `Router(config)# controller sonet 0/0/0`	Returns to controller configuration mode.
Step 2	**aps group** *group-number* **acr** **Example:** `Router(config-if)# aps group acr 1`	Configures the working interface group on a router. The APS group number must be greater than 1.
Step 3	**aps working** *circuit-number* **Example:** `Router(config-if)# aps working 1`	Configures this interface as a working interface. 1 is the only supported *circuit-number* value.

	Command or Action	Purpose
Step 4	**end** **Example:** `Router(config-if)# end`	Exits configuration mode.

Configuring a Protect Interface

To configure a protect interface, use the following commands beginning in global configuration mode.

Before you begin

To configure the controller in SDH mode, see Configuring Optical Interface Modules.

Procedure

	Command or Action	Purpose
Step 1	**controller sonet** *slot* / *port-adapter* / *port* **Example:** `Router(config)# controller sonet 0/0/0`	Returns to controller configuration mode.
Step 2	**aps group** *group-number* **acr** **Example:** `Router(config-if)# aps group acr 2`	(Optional) Allows more than one protect/working interface group to be supported on a router.
Step 3	**aps protect** *circuit-number ip-address* **Example:** `Router(config-if)# aps protect 1 7.7.7.7`	Configures the interface as a protect interface and specifies the IP address of the device that contains the working interface.
Step 4	**end** **Example:** `Router(config-if)# end`	Exits configuration mode.

Configuring Other APS Options

To configure the other APS options, use any of the following optional commands in interface configuration mode.

Procedure

	Command or Action	Purpose
Step 1	**aps authenticate** *string* **Example:** `Router(config-if)# aps authenticate authstring`	(Optional) Configures the authentication string that the router uses to authenticate PGP message exchange between protect or working routers. The maximum length of the string is eight alphanumeric characters. Spaces are not accepted.
Step 2	**aps force** *circuit-number* **Example:** `Router(config-if)# aps force 1`	(Optional) Manually switches the specified circuit to a protect interface, unless a request of equal or higher priority is in effect. For example, if the protect interface is configured as circuit 1, use the **aps force 1** command to set the protect interface to active. **Note** If you do not want the protect port to be active all the time, use **no aps force 1** command after using **aps force 1** command. Similarly for **aps force 0** use use **no aps force 0** command.
Step 3	**aps group** *group-number* **Example:** `Router(config-if)# aps group 2`	(Optional) Allows more than one protect/working interface group to be supported on a router.
Step 4	**aps lockout** *circuit-number* **Example:** `Router(config-if)# aps lockout 1`	(Optional) Prevents a working interface from switching to a protect interface. For example, if the protect interface is configured as circuit 1, use the **aps lockout 1** command to prevent the protect interface from becoming active.
Step 5	**aps manual** *circuit-number* **Example:** `Router(config-if)# aps manual 0`	(Optional) Manually switches a circuit to a protect interface, unless a request of equal or higher priority is in effect. For example, if the working interface is configured as circuit 0, the command is applied as follows: • The **aps manual 0** command activates the working interface • The **aps manual 1** command activates the protect circuit. Applying the **no** form of the command removes the configuration and stops the router from sending K 1and K2 bytes on the interface.

	Command or Action	Purpose
Step 6	**aps revert** *minutes* **Example:** `Router(config-if)# aps revert 10`	(Optional) Enables automatic switchover from the protect interface to the working interface after the working interface becomes available.
Step 7	**aps timers** *seconds1 seconds2* **Example:** `Router(config-if)# aps timers 1 5`	(Optional) Specifies the following values: • *seconds1*—The time between hello packets. • *seconds2*—The time that the working interface can be down before the router switches to the protect interface.
Step 8	**aps unidirectional** **Example:** `Router(config-if)# aps unidirectional`	(Optional) Configures a protect interface for unidirectional mode.

Example

```
Router# configure terminal
Router# interface gigabit ethernet 0/1/0
Router(config-if)# aps force 1
Ruter(config-if)# aps unidirectional
```

Stateful MLPPP Configuration with MR-APS Inter-Chassis Redundancy

The Cisco ASR 903 Router supports Stateful MLPPP with Inter-Chassis Redundancy. For information on how to configure this feature, see http://www.cisco.com/en/US/docs/ios/wan/configuration/guide/wan_mlppp_mr_aps.html.

Monitoring and Maintaining APS

To provide information about system processes, the Cisco IOS software includes an extensive list of EXEC commands that begin with the word show, which, when executed, display detailed tables of system information. Following is a list of some of the common show commands for the APS feature.

To display the information described, use these commands in privileged EXEC mode.

- Use the **show aps** command to display information about APS.
- Use the **show controller sonet** *slot* command to display information about the controller port.

- use the **show interfaces** command to display information about the interface.

For more information about these commands, see the *Cisco IOS Interface and Hardware Component Command Reference*.

CHAPTER 3

Configuring Multi Router Automatic Protection Switching

Note Multi Router Automatic Protection Switching is *not* supported on the Cisco ASR 900 RSP3 module.

The Multi Router Automatic Protection Switching (MR-APS) integration with hot standby pseudowire (HSPW) feature is a protection mechanism for Synchronous Optical Network (SONET) networks that enables SONET connections to switch to another SONET circuit when a circuit failure occurs. A protect interface serves as the backup interface for the working interface. When the working interface fails, the protect interface quickly assumes its traffic load.

Note When you perform protect-active router powercycle, the convergence times becomes high ranging from 2.3 sconds to 2.8 seconds. The APS switchover triggers the PWs at the protect interface to become active during any one of the following failure scenarios:

- Either port at the ADM does not respond.
- The port at the router does not respond.
- The link between ADM and router fails.
- The router fails over.

Restrictions for MR-APS

- Asynchronous Transfer Mode (ATM) port mode is not supported.
- An APS group number must be greater than zero.

- Revertive APS mode on the Circuit Emulation (CEM) interface is not supported.
- Starrting with Cisco IOS XE Release 3.15, CEM MR-APS switvchover does not occur on an RP SSO.
- HSPW *group number* other than the redundancy interchassis *group number* is not supported.
- Do not configure the **backup delay** *value* command if the MR-APS integration with HSPW feature is configured.
- Unconfiguring the **mpls ip** command on the core interface is not supported.
- The **hspw force switch** command is not supported.
- When you enable MRAPS 1+1 unidirectional mode, the PW status does not change for ASR 903 routers. But, the same behavior is not seen for ASR 901 routers. To overcome this issue, reload the ASR 901 router.
- Ensure to have both ASR 903 and ASR 901 routers configured with unidirectional configuration mode for MRAPS 1+1, else it results in a traffic drop.

Information About MR-APS

This feature enables interface connections to switch from one circuit to another if a circuit fails. Interfaces can be switched in response to a router failure, degradation or loss of channel signal, or manual intervention. In a multi router environment, the MR-APS allows the protected SONET interface to reside in a different router from the working SONET interface.

Service providers are migrating to ethernet networks from their existing SONET or SDH equipment to reduce cost. Any transport over MPLS (AToM) PWs help service providers to maintain their investment in time division multiplexing (TDM) network and change only the core from SONET or SDH to ethernet. When the service providers move from SONET or SDH to ethernet, network availability is always a concern. Therefore, to enhance the network availability, service providers use PWs.

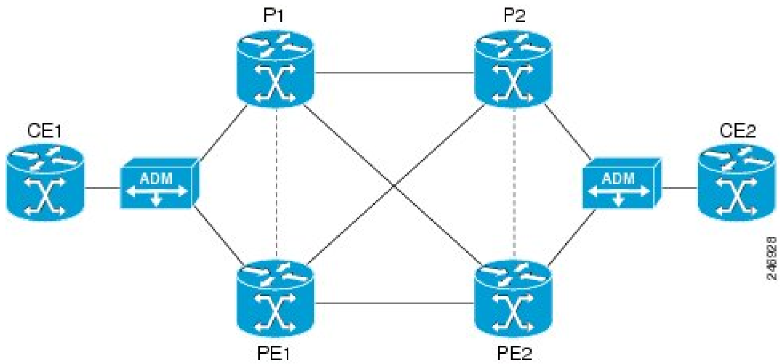
The HSPW support for TDM access circuits (ACs) allow the backup PW to be in a hot- standby state, so that it can immediately take over if the primary PW fails. The present HSPW solution does not support ACs as part of the APS group. The PWs which are configured over the protected interface, remain in the standby state. MR-APS integration with an HSPW is an integration of APS with CEM TDM HSPW and improves the switchover time.

For more information on APS, see the Automatic Protection Switching Configuration.

In the example below, routers P1 and PE1 are in the same APS group G1, and routers P2 and PE2 are in the same APS group G2. In group G1, P1 is the working router and PE1 is the protected router. Similarly in group G2, P2 is the working router and PE2 is the protected router.

The MR-APS integration with HSPW deployment involves cell sites connected to the provider network using bundled T1/E1 connections. These T1/E1 connections are aggregated into the optical carrier 3 (OC3) link using the add-drop multiplexers (ADMs).

Figure 6: MR-APS Integration with HSPW Implementation

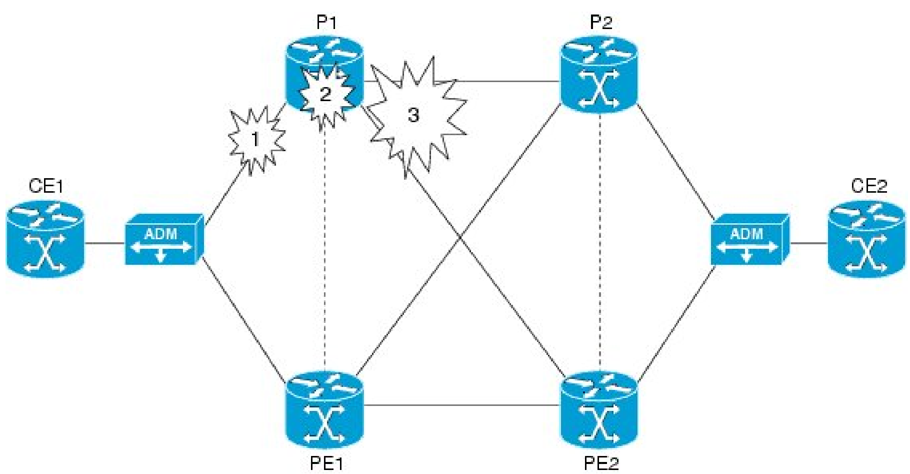


Failover Operations

MR-APS integration with HSPW feature handles the following failures:

- Failure 1, where the link between ADM and P1 goes down, or the connecting ports at ADM or P1 go down.
- Failure 2, where the router P1 fails.
- Failure 3, where the router P1 is isolated from the core.

Figure 7: Failure Points in the Network

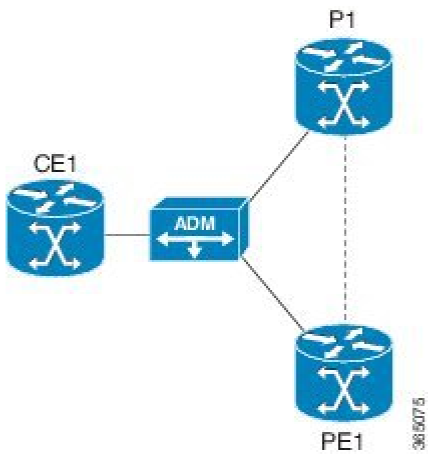


In case of failure 1, where either port at the ADM goes down, or the port at the router goes down, or the link between ADM and router fails, the APS switchover triggers the pseudowires at the protect interface to become active. The same applies to failure 2 as well where the complete router fails over.

In case of failure 3, where all the links carrying primary and backup traffic lose the connection, a new client is added to the inter chassis redundancy manager (ICRM) infrastructure to handle the core isolation. The client listens to the events from the ICRM. Upon receiving the core isolation event from the ICRM, the client either initiates the APS switchover, or initiates the alarm based on the peer core isolation state. If APS switchover occurs, it changes the APS inactive interface to active and hence activates the PWs at the interface. Similarly, when core connectivity goes up based upon the peer core isolation state, it clears the alarms or triggers the

APS switchover. The ICRM monitors the directly connected interfaces only. Hence only those failures in the directly connected interfaces can cause a core isolation event.

Figure 8: MR-APS Integration on a POS interface



Configuring MR-APS with HSPW-ICRM on a CEM interface

To configure MR-APS integration with HSPW-ICRM on a CEM interface, complete the following steps:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** *pw-class-name* **Example:** `Router(config)# pseudowire-class hspw_aps`	Specifies the name of a PW class and enters PW class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies that MPLS is used as the data encapsulation method for tunneling Layer 2 traffic over the PW.
Step 5	**status peer topology dual-homed** **Example:**	Enables the reflection of the attachment circuit status on both the primary and secondary PWs.

	Command or Action	Purpose
	Router(config-pw-class)# **status peer topology dual-homed**	This configuration is necessary if the peer PEs are connected to a dual-homed device.
Step 6	**exit** **Example:** Router(config-pw-class)# **exit**	Exits PW class configuration mode.
Step 7	**redundancy** **Example:** Router(config)# **redundancy**	Enters the redundancy configuration mode.
Step 8	**interchassis group** *group-id* **Example:** Router(config-red)# **interchassis group 50**	Configures an interchassis group within the redundancy configuration mode and enters the interchassis redundancy mode.
Step 9	**member ip** *ip-address* **Example:** Router(config-r-ic)# **member ip 60.60.60.2**	Configures the IP address of the peer member group.
Step 10	**backbone interface** *slot/bay/port* **Example:** Router(config-r-ic)# **backbone interface GigabitEthernet 0/2/3**	Specifies the backbone interface. • *slot*—Chassis slot number, which is always 0. • *bay*—Card interface bay number in a slot. The range is from 0 to 5. • *port*—Port or interface number. The range is from 0 to 7 for Gigabit Ethernet.
Step 11	**exit** **Example:** Router(config-r-ic)# **exit**	Exits the redundancy mode.
Step 12	**controller SONET** *slot/bay/port* **Example:** Router(config)# **controller SONET 0/5/2**	Selects and configures a SONET controller and enters controller configuration mode. • *slot*—Chassis slot number, which is always 0. • *bay*—Card interface bay number in a slot. The range is from 0 to 5. • *port*—Port or interface number. The range is from 0 to 7 for Gigabit Ethernet.

	Command or Action	Purpose
Step 13	**framing** [**SDH** | **SONET**] **Example:** `Router(config-controller)# framing SONET`	Configures the controller with framing type. SONET framing is the default option.
Step 14	**clock source line** **Example:** `Router(config-controller)# clock source line`	Sets the clocking for individual T1 or E1 links.
Step 15	**sts-1** *sts1-number* **Example:** `Router(config-controller)# sts-1 1`	Specifies the STS identifier.
Step 16	**mode vt-15** **Example:** `Router(config-ctrlr-sts1)# mode vt-15`	Specifies the STS-1 mode of operation.
Step 17	**vtg** *vtg_number* **t1** *t1_line_number* **cem-group** *group-number* **timeslots** *time-slot-range* **Example:** `Router(config-ctrlr-sts1)# vtg 1 t1 1 cem-group 0 timeslots 1-24`	Creates a Circuit Emulation Services over Packet Switched Network circuit emulation (CESoPSN) CEM group. • **vtg**—Specifies the VTG number from 1-7. • **t1**—Specifies the T1 line. • *t1_line_number*—Specifies the T1 line number. • **cem-group**—Creates a circuit emulation (CEM) channel from one or more time slots of a T1 line. • *group-number*—CEM identifier to be used for this group of time slots. For T1 ports, the range is from 0 to 23. • **timeslots**—Specifies that a list of time slots is to be used as specified by the *time-slot-range* argument. • *time-slot-range*—Specifies the time slots to be included in the CEM channel. The list of time slots may include commas and hyphens with no spaces between the numbers.
Step 18	**exit** **Example:** `Router(config-ctrlr-sts1)# exit`	Exits from the STS configuration mode.

	Command or Action	Purpose
Step 19	**aps group** *group_id* **Example:** `Router(config-controller)# aps group 1`	Configures the APS group for CEM.
Step 20	**aps** [**working** | **protect**] *aps-group-number* **Example:** `Router(config-controller)# aps working 1`	Configures the APS group as working or protect interface. **Note** For MR-APS, one router must be configured as aps working 1 and the other router must be configured as aps protect 1.
Step 21	**aps hspw-icrm-grp** *group-number* **Example:** `Router(config-controller)# aps hspw-icrm-group 1`	Associates the APS group to an ICRM group number.
Step 22	**exit** **Example:** `Router(config-controller)# exit`	Ends the controller session and returns to the configuration mode.
Step 23	**interface cem** *slot/bay/port* **Example:** `Router(config)# interface cem 0/5/2`	Configures a serial interface and enters the interface configuration mode • *slot*—Chassis slot number, which is always 0. • *bay*—Card interface bay number in a slot. The range is from 0 to 5. • *port*—Port or interface number. The range is from 0 to 7 for Gigabit Ethernet.
Step 24	**cem** *group-number* **Example:** `Router(config-if)# cem 0`	Selects the CEM circuit (group) to configure a PW for.
Step 25	**xconnect** *peer-ip-address vcid* **pw-class** *pw-class-name* **Example:** `Router(config-if-srv)# xconnect 3.3.3.3 1 pw-class hspw_aps`	Specifies the IP address of the peer PE router and the 32-bit virtual circuit identifier shared between the PEs at each end of the control channel. • *peer-ip-address*—IP address of the remote provider edge (PE) peer. The remote router ID can be any IP address, as long as it is reachable. • *vcid* —32-bit identifier of the virtual circuit (VC) between the PE routers.

	Command or Action	Purpose
		• **pw-class**—Specifies the PW class. • *pw-class-name*—Specifies the name of the PW class. **Note** The peer router IP address and virtual circuit ID must be a unique combination on the router.
Step 26	**backup peer** *peer-id vc-id* **pw-class** *pw-class-name* **Example:** `Router(config-if-srv)# backup peer 4.3.3.3 90 pw-class vpws`	Specifies a redundant peer for a PW virtual circuit. • *peer-id vc-id*—Specifies IP address of the remote peer. • **pw-class**—Specifies the PW class. • *pw-class-name*—Specifies the name of the PW class.
Step 27	**end** **Example:** `Router(config-if-srv)# end`	Returns to privileged EXEC mode.

Verifying MR-APS

- Use the **show cem circuit** [*cem-group-id* | **interface** {**CEM** | **Virtual-CEM**} *slot /subslot /port cem-group-id* | **detail** | **summary**] command to display CEM statistics for the configured CEM circuits. If **xconnect** is configured under the circuit, the command output also includes information about the attached circuit.

  Following is a sample output of the **show cem circuit** command to display the detailed information about CEM circuits configured on the router:

  `Router# show cem circuit`

```
CEM Int.        ID   Ctrlr     Admin     Circuit         AC
--------------------------------------------------------------
CEM0/5/2        1    UP        UP        Active           UP
CEM0/5/2        2    UP        UP        Active           UP
CEM0/5/2        3    UP        UP        Active           UP

!
.
.
.

CEM0/5/2        83   UP        UP        Active           UP
CEM0/5/2        84   UP        UP        Active           UP
!
```

Following is a sample output of the **show cem circuit***0-504* command to display the detailed information about that particular circuit:

```
Router# show cem circuit 1
```

```
CEM0/5/2 , ID: 1, Line: UP, Admin: UP, Ckt: ACTIVE Controller state: up, T1/E1
 state: up Idle Pattern: 0xFF, Idle CAS: 0x8
Dejitter: 5 (In use: 0)
Payload Size: 192
Framing: Unframed
CEM Defects Set
None

Signalling: No CAS
RTP: No RTP

Ingress Pkts:    151066               Dropped:             0

Egress Pkts:     151066               Dropped:             0


CEM Counter Details
Input Errors:    0                    Output Errors:       0

Pkts Missing:    0                    Pkts Reordered:      0

Misorder Drops:  0                    JitterBuf Underrun:  0

Error Sec:       0                    Severly Errored Sec: 0

Unavailable Sec: 0                    Failure Counts:      0

Pkts Malformed:  0                    JitterBuf Overrun:   0
```

- Use the **show mpls ldp neighbor** command to display the status of Label Distribution Protocol (LDP) sessions:

```
Router# show mpls ldp neighbor
```

```
 Peer LDP Ident: 17.3.3.3:0; Local LDP Ident 17.1.1.1:0
        TCP connection: 17.3.3.3.13282 - 17.1.1.1.646
        State: Oper; Msgs sent/rcvd: 466/209; Downstream
        Up time: 00:23:50
        LDP discovery sources:
          GigabitEthernet0/4/0 , Src IP addr: 11.11.11.2
          Targeted Hello 17.1.1.1 -> 17.3.3.3, active, passive
        Addresses bound to peer LDP Ident:
          70.70.70.1      22.22.22.2      17.3.3.3        11.11.11.2
    Peer LDP Ident: 17.4.4.4:0; Local LDP Ident 17.1.1.1:0
        TCP connection: 17.4.4.4.24248 - 17.1.1.1.646
        State: Oper; Msgs sent/rcvd: 209/205; Downstream
        Up time: 00:23:40
        LDP discovery sources:
          GigabitEthernet0/4/2, Src IP addr: 33.33.33.2
          Targeted Hello 17.1.1.1 -> 17.4.4.4, active, passive
        Addresses bound to peer LDP Ident:
          70.70.70.2      44.44.44.2      17.4.4.4        33.33.33.2
    Peer LDP Ident: 17.2.2.2:0; Local LDP Ident 17.1.1.1:0
        TCP connection: 17.2.2.2.32112 - 17.1.1.1.646
        State: Oper; Msgs sent/rcvd: 45/44; Downstream
        Up time: 00:23:38
        LDP discovery sources:
```

```
          GigabitEthernet0/4/4 , Src IP addr: 60.60.60.2
        Addresses bound to peer LDP Ident:
          22.22.22.1      44.44.44.1      17.2.2.2        60.60.60.2
```

- Use the **show mpls l2 vc** command to display information related to a VC:

```
Router# show mpls l2 vc
```

```
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ----------
----------
CEM0/5/2           SATOP T1 1                 17.3.3.3        1001       UP

CEM0/5/2           SATOP T1 2                 17.3.3.3        1002       UP

CEM0/5/2           SATOP T1 3                 17.3.3.3        1003       UP
!
.
.
.

CEM0/5/2           SATOP T1 19                17.3.3.3        1019       UP

CEM0/5/2           SATOP T1 20                17.3.3.3        1020       UP
!

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ----------
----------
CEM0/5/2           SATOP T1 21                17.3.3.3        1021       UP

CEM0/5/2           SATOP T1 22                17.3.3.3        1022       UP

CEM0/5/2           SATOP T1 23                17.3.3.3        1023       UP

!
.
.
.

CEM0/5/2           SATOP T1 25                17.3.3.3        1025       UP

CEM0/5/2           SATOP T1 43                17.3.3.3        1043       UP
!

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ----------
----------
CEM0/5/2           SATOP T1 44                17.3.3.3        1044       UP

CEM0/5/2           SATOP T1 45                17.3.3.3        1045       UP

CEM0/5/2           SATOP T1 46                17.3.3.3        1046       UP

!
.
.

CEM0/5/2           SATOP T1 65                17.3.3.3        1065       UP

CEM0/5/2           SATOP T1 66                17.3.3.3        1066       UP
```

```
!
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ----------
----------
CEM0/5/2           SATOP T1 67                17.3.3.3        1067       UP

CEM0/5/2           SATOP T1 68                17.3.3.3        1068       UP

CEM0/5/2           SATOP T1 69                17.3.3.3        1069       UP

!
.
.
.

CEM0/5/2           SATOP T1 83                17.3.3.3        1083       UP

CEM0/5/2           SATOP T1 84                17.3.3.3        1084       UP

CEM0/5/2           SATOP T1 1                 17.4.4.4        4001
STANDBY
CEM0/5/2           SATOP T1 2                 17.4.4.4        4002
STANDBY
CEM0/5/2           SATOP T1 3                 17.4.4.4        4003
STANDBY
CEM0/5/2           SATOP T1 4                 17.4.4.4        4004
STANDBY
CEM0/5/2           SATOP T1 5                 17.4.4.4        4005
STANDBY

!
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ----------
----------
CEM0/5/2           SATOP T1 6                 17.4.4.4        4006
STANDBY
CEM0/5/2           SATOP T1 7                 17.4.4.4        4007
STANDBY
CEM0/5/2           SATOP T1 8                 17.4.4.4        4008
STANDBY

!
.
.
.

CEM0/5/2           SATOP T1 27                17.4.4.4        4027
STANDBY
CEM0/5/2           SATOP T1 28                17.4.4.4        4028
STANDBY

!
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ----------
----------
CEM0/5/2           SATOP T1 29                17.4.4.4        4029
STANDBY
CEM0/5/2           SATOP T1 30                17.4.4.4        4030
STANDBY
CEM0/5/2           SATOP T1 31                17.4.4.4        4031
STANDBY
```

```
!
.
.
.
CEM0/5/2          SATOP T1 50                17.4.4.4       4050
STANDBY
CEM0/5/2          SATOP T1 51                17.4.4.4       4051
STANDBY

!

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ----------
----------
CEM0/5/2          SATOP T1 52                17.4.4.4       4052
STANDBY
CEM0/5/2          SATOP T1 53                17.4.4.4       4053
STANDBY
CEM0/5/2          SATOP T1 54                17.4.4.4       4054
STANDBY

!
.
.
.
CEM0/5/2          SATOP T1 73                17.4.4.4       4073
STANDBY
CEM0/5/2          SATOP T1 74                17.4.4.4       4074
STANDBY

!

Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ----------
----------
CEM0/5/2          SATOP T1 75                17.4.4.4       4075
STANDBY
CEM0/5/2          SATOP T1 76                17.4.4.4       4076
STANDBY
CEM0/5/2          SATOP T1 77                17.4.4.4       4077
STANDBY

!
.
.
.
CEM0/5/2          SATOP T1 83                17.4.4.4       4083
STANDBY
CEM0/5/2          SATOP T1 84                17.4.4.4       4084
STANDBY

!

R-96-2011#sh cem circuit
CEM Int.       ID   Ctrlr     Admin     Circuit         AC
--------------------------------------------------------------
CEM0/5/2         1    UP        UP        Active           UP
CEM0/5/2         2    UP        UP        Active           UP
CEM0/5/2         3    UP        UP        Active           UP
```

```
!
.
.
.

CEM0/5/2          83   UP        UP        Active           UP
CEM0/5/2          84   UP        UP        Active           UP

!
```

- Use the **show mpls l2 vc** *vc-id* **detail** command to display detailed information related to the VC:

```
Router# show mpls l2 vc 1001 detail
```

```
Local interface: CEM0/5/2    up, line protocol up, SATOP T1 1 up
  Destination address: 17.3.3.3, VC ID: 1001, VC status: up
    Output interface: Gi0/4/0 , imposed label stack {42}
    Preferred path: not configured
    Default path: active
    Next hop: 11.11.11.2
  Create time: 00:26:04, last status change time: 00:03:36
    Last label FSM state change time: 00:23:00
  Signaling protocol: LDP, peer 17.3.3.3:0 up
    Targeted Hello: 17.1.1.1(LDP Id) -> 17.3.3.3, LDP is UP
    Graceful restart: configured and enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 182, remote 42
    Group ID: local 0, remote 0
    MTU: local 0, remote 0
    Remote interface description:
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense)
  SSO Descriptor: 17.3.3.3/1001, local label: 182
  Dataplane:
    SSM segment/switch IDs: 1278679/4262 (used), PWID: 1
  VC statistics:
    transit packet totals: receive 201616, send 201617
    transit byte totals:   receive 41129664, send 40323400
    transit packet drops:  receive 0, seq error 0, send 0
```

- Use the **show hspw-aps-icrm group** *group-id* command to display information about a specified HSPW APS group:

```
Router# show hspw-aps-icrm group 100
```

```
ICRM group id 100, Flags : My core isolated No,Peer core isolated No, State
Connect
```

```
         APS Group id 1 hw_if_index 33 APS valid:Yes
        Total aps grp attached to ICRM group 100 is 1
```

- Use the **show hspw-aps-icrm all** command to display information about all HSPW APS and ICRM groups:

```
Router# show hspw-aps-icrm all
```

```
ICRM group id 100, Flags : My core isolated No,Peer core isolated No, State
Connect
         APS Group id 1 hw_if_index 33 APS valid:Yes
       Total aps grp attached to ICRM group 100 is 1 ICRM group count attached
 to MR-APS HSPW feature is 1
```

- Use the **show redundancy interchassis** command to display information about interchassis redundancy group configuration:

```
Router# show redundancy interchassis
```

```
Redundancy Group 100 (0x64)
  Applications connected: MR-APS with HSPW
  Monitor mode: RW
  member ip: 60.60.60.2 "R-222-2028", CONNECTED
    Route-watch for 60.60.60.2 is UP
    MR-APS with HSPW state: CONNECTED
  backbone int GigabitEthernet0/4/0 : UP (IP)
  backbone int GigabitEthernet0/4/2 : UP (IP)

ICRM fast-failure detection neighbor table
  IP Address       Status Type Next-hop IP      Interface
  ==========       ====== ==== ===========      =========
  60.60.60.2       UP     RW
```

- Use the **show aps** command to display information about the current APS feature:

```
Router# show aps
```

```
 SONET 0/5/2    APS Group 1: working channel 1 (Active) (HA)
        Protect at 60.60.60.2
        PGP timers (from protect): hello time=1; hold time=10
        SONET framing
        Remote APS configuration: (null)
```

- Use the **show xconnect all** command to display information about all Cross–Connect attachment circuits and PWs:

```
Router# show xconnect all
```

```
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2
   S2
------+---------------------------------+--+----------------------------
------+---------------------------------+--+---
--+--
UP pri   ac CEM0/5/2 :1(SATOP T1)          UP mpls 17.3.3.3:1001
   UP
```

```
IA sec   ac CEM0/5/2 :1(SATOP T1)          UP mpls 17.4.4.4:4001
   SB
UP pri   ac CEM0/5/2 :10(SATOP T1)         UP mpls 17.3.3.3:1010
   UP
IA sec   ac CEM0/5/2 :10(SATOP T1)         UP mpls 17.4.4.4:4010
   SB

!
.
.
.

UP pri   ac CEM0/5/2 :9(SATOP T1)          UP mpls 17.3.3.3:1009

   UP
IA sec   ac CEM0/5/2 :9(SATOP T1)          UP mpls 17.4.4.4:4009

   SB

!
```

Configuration Examples for MR-APS

The following example shows how to configure the MR-APS integration with HSPW on a CEM interface on the working router with framing mode as SONET on router P1:

```
RouterP1> enable
RouterP1# configure terminal
RouterP1(config)# pseudowire-class hspw_aps
RouterP1(config-pw-class)# encapsulation mpls
RouterP1(config-pw-class)# status peer topology dual-homed
RouterP1(config-pw-class)# exit
RouterP1(config)# redundancy
RouterP1(config-red)# interchassis group 1
RouterP1(config-r-ic)# member ip 14.2.0.2
RouterP1(config-r-ic)# backbone interface GigabitEthernet 0/1/0
RouterP1(config-r-ic)# backbone interface GigabitEthernet 0/1/1
RouterP1(config-r-ic)# exit
RouterP1(config)# controller SONET 0/1/0
RouterP1(config-controller)# framing sonet
RouterP1(config-controller)# clock source line
RouterP1(config-controller)# sts-1 1
RouterP1(config-ctrlr-sts1)# mode vt-15
RouterP1(config-ctrlr-sts1)# vtg 1 t1 1 cem-group 0 timeslots 1-24
RouterP1(config-ctrlr-sts1)# exit
RouterP1(config-controller)# aps group 3
RouterP1(config-controller)# aps working 1
RouterP1(config-controller)# aps hspw-icrm-grp 1
RouterP1(config-controller)# exit
RouterP1(config)# interface cem 0/1/0
RouterP1(config-if)# cem 0
RouterP1(config-if)# xconnect 3.3.3.3 1 encapsulation mpls pw-class hspw_aps
RouterP1(config-if)# backup peer 4.4.4.4 2 pw-class hspw_aps
RouterP1(config-if)# exit
RouterP1(config)# end
```

The following example shows how to configure the MR-APS integration with HSPW on a CEM interface on the protect router with framing mode as SONET on router PE1:

```
RouterPE1> enable
RouterPE1# configure terminal
```

```
RouterPE1(config)# pseudowire-class hspw_aps
RouterPE1(config-pw-class)# encapsulation mpls
RouterPE1(config-pw-class)# status peer topology dual-homed
RouterPE1(config-pw-class)# exit
RouterPE1(config)# redundancy
RouterPE1(config-red)# interchassis group 1
RouterPE1(config-r-ic)# member ip 14.2.0.1
RouterPE1(config-r-ic)# backbone interface GigabitEthernet 0/1/0
RouterPE1(config-r-ic)# backbone interface GigabitEthernet 0/1/1
RouterPE1(config-r-ic)# exit
RouterPE1(config)# controller SONET 0/2/0
RouterPE1(config-controller)# framing sonet
RouterPE1(config-controller)# clock source line
RouterPE1(config-controller)# sts-1 1
RouterPE1(config-ctrlr-sts1)# mode vt-15
RouterPE1(config-ctrlr-sts1)# vtg 1 t1 1 cem-group 0 timeslots 1-24
RouterPE1(config-ctrlr-sts1)# exit
RouterPE1(config-controller)# aps group 3
RouterPE1(config-controller)# aps protect 1 14.2.0.2
RouterPE1(config-controller)# aps hspw-icrm-grp 1
RouterPE1(config-controller)# exit
RouterPE1(config)# interface cem 0/2/0
RouterPE1(config-if)# cem 0
RouterPE1(config-if)# xconnect 3.3.3.3 3 pw-class hspw_aps
RouterPE1(config-if)# backup peer 4.4.4.4 4 pw-class hspw_aps
RouterPE1(config-if)# exit
RouterPE1(config)# end
```

The following example shows how to configure the MR-APS integration with HSPW on a CEM interface on the working router with framing mode as SONET on router P2:

```
RouterP2> enable
RouterP2# configure terminal
RouterP2(config)# pseudowire-class hspw_aps
RouterP2(config-pw-class)# encapsulation mpls
RouterP2(config-pw-class)# status peer topology dual-homed
RouterP2(config-pw-class)# exit
RouterP2(config)# redundancy
RouterP2(config-red)# interchassis group 1
RouterP2(config-r-ic)# member ip 14.6.0.2
RouterP2(config-r-ic)# backbone interface GigabitEthernet 0/2/0
RouterP2(config-r-ic)# backbone interface GigabitEthernet 0/2/1
RouterP2(config-r-ic)# exit
RouterP2(config)# controller SONET 0/1/0
RouterP2(config-controller)# framing sonet
RouterP2(config-controller)# clock source line
RouterP2(config-controller)# sts-1 1
RouterP2(config-ctrlr-sts1)# mode vt-15
RouterP2(config-ctrlr-sts1)# vtg 1 t1 1 cem-group 0 timeslots 1-24
RouterP2(config-ctrlr-sts1)# exit
RouterP2(config-controller)# aps group 3
RouterP2(config-controller)# aps working 1
RouterP2(config-controller)# aps hspw-icrm-grp 1
RouterP2(config-controller)# exit
RouterP2(config)# interface cem 0/1/0
RouterP2(config-if)# cem 0
RouterP2(config-if)# xconnect 10.1.1.1 1 encapsulation mpls pw-class hspw_aps
RouterP2(config-if)# backup peer 2.2.2.2 3 pw-class hspw_aps
RouterP2(config-if)# exit
RouterP2(config)# end
```

The following example shows how to configure the MR-APS Integration with HSPW on a CEM interface on the protect router with framing mode as SONET on router PE2:

```
RouterPE2> enable
RouterPE2# configure terminal
RouterPE2(config)# pseudowire-class hspw_aps
RouterPE2(config-pw-class)# encapsulation mpls
RouterPE2(config-pw-class)# status peer topology dual-homed
RouterPE2(config-pw-class)# exit
RouterPE2(config)# redundancy
RouterPE2(config-red)# interchassis group 1
RouterPE2(config-r-ic)# member ip 14.6.0.1
RouterPE2(config-r-ic)# backbone interface GigabitEthernet 0/2/0
RouterPE2(config-r-ic)# backbone interface GigabitEthernet 0/2/1
RouterPE2(config-r-ic)# exit
RouterPE2(config)# controller SONET 0/2/0
RouterPE2(config-controller)# framing sonet
RouterPE2(config-controller)# clock source line
RouterPE2(config-controller)# sts-1 1
RouterPE2(config-ctrlr-sts1)# mode vt-15
RouterPE2(config-ctrlr-sts1)# vtg 1 t1 1 cem-group 0 timeslots 1-24
RouterPE2(config-ctrlr-sts1)# exit
RouterPE2(config-controller)# aps group 2
RouterPE2(config-controller)# aps protect 1 14.6.0.2
RouterPE2(config-controller)# aps hspw-icrm-grp 1
RouterPE2(config-controller)# exit
RouterPE2(config)# interface cem 0/2/0
RouterPE2(config-if)# cem 0
RouterPE2(config-if)# xconnect 10.1.1.1 2 pw-class hspw_aps
RouterPE2(config-if)# backup peer 2.2.2.2 4 pw-class hspw_aps
RouterPE2(config-if)# exit
RouterPE2(config)# end
```

Configuring MR-APS on a POS interface

The following section shows how to configure the MR-APS integration on a POS interface on the working node and protect node.

Configuring working node for POS MR-APS

To configure MR-APS working node for POS interface, complete the following steps:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**exit** **Example:** `Router(config-pw-class)# exit`	Exits PW class configuration mode.

	Command or Action	Purpose
Step 4	**redundancy** **Example:** `Router(config)# redundancy`	Enters the redundancy configuration mode.
Step 5	**interchassis group** *group-id* **Example:** `Router(config-red)# interchassis group 50`	Configures an interchassis group within the redundancy configuration mode and enters the interchassis redundancy mode.
Step 6	**member ip** *ip-address* **Example:** `Router(config-r-ic)# member ip 60.60.60.2`	Configures the IP address of the peer member group.
Step 7	**monitor peer** *bfd* **Example:** `Router(config-red)# monitor peer bfd`	Enables BFD on the POS link.
Step 8	**exit** **Example:** `Router(config-r-ic)# exit`	Exits the redundancy mode.
Step 9	**controller SONET** *slot/bay/port* **Example:** `Router(config)# controller SONET 0/5/2`	Selects and configures a SONET controller and enters controller configuration mode. • *slot*—Chassis slot number, which is always 0. • *bay*—Card interface bay number in a slot. The range is from 0 to 5. • *port*—Port or interface number. The range is from 0 to 7 for Gigabit Ethernet.
Step 10	**framing** [**SDH** | **SONET**] **Example:** `Router(config-controller)# framing SONET`	Configures the controller with framing type. SONET framing is the default option.
Step 11	**clock source internal** **Example:** `Router(config-controller)# clock source internal`	Sets the clocking for individual E1 links.
Step 12	**sts-1** *1-3POS* **Example:** `Router(config-controller)# sts-1 1-3`	Specifies the STS identifier.

	Command or Action	Purpose
Step 13	**exit** **Example:** `Router(config-ctrlr-sts1)# exit`	Exits from the STS configuration mode.
Step 14	**controller SONET** *slot/bay/port* **Example:** `Router(config)# controller SONET 0/5/2`	Selects and configures a SONET controller and enters controller configuration mode.
Step 15	**Shutdown** **Example:** `Router(config)# Shutdown`	Shut down the controller before APS configuration.
Step 16	**aps group** *group_id* **Example:** `Router(config-controller)# aps group 1`	Configures the APS group for POS.
Step 17	**aps working** *aps-group-number* **Example:** `Router(config-controller)# aps working 1`	Configures the APS group as working or protect interface. **Note** For MR-APS, one router must be configured as aps working 1 and the other router must be configured as aps protect 1.
Step 18	**aps interchassis group** *group-id* **Example:** `Router(config-red)# aps interchassis group 50`	Configures an aps inter chassis group.
Step 19	**no shut** **Example:** `Router(config-controller)# no shut`	Shut down the controller.
Step 20	**exit** **Example:** `Router(config-controller)# exit`	Ends the controller session and returns to the configuration mode.
Step 21	**interface POS** *slot/bay/port* **Example:** `Router(config)# interface POS 0/5/2`	Configures a serial interface and enters the interface configuration mode • *slot*—Chassis slot number, which is always 0. • *bay*—Card interface bay number in a slot. The range is from 0 to 5. • *port*—Port or interface number. The range can be 0-3.

	Command or Action	Purpose
Step 22	**ip address** *ip-address* **Example:** `Router(config-if)# ip address 45.1.1.2 255.255.255.0`	Assigns the ip address to POS interface
Step 23	**encapsulation** *ppp* **Example:** `Router(config-if-srv)# encapsulation ppp`	Specifies the ppp encapsulation over POS interface.
Step 24	**end** **Example:** `Router(config-if-srv)# end`	Returns to privileged EXEC mode.

Configuring protect node for POS MR-APS

To configure MR-APS protect node for POS interface, complete the following steps:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**exit** **Example:** `Router(config-pw-class)# exit`	Exits PW class configuration mode.
Step 4	**redundancy** **Example:** `Router(config)# redundancy`	Enters the redundancy configuration mode.
Step 5	**interchassis group** *group-id* **Example:** `Router(config-red)# interchassis group 50`	Configures an interchassis group within the redundancy configuration mode and enters the interchassis redundancy mode.
Step 6	**member ip** *ip-address* **Example:**	Configures the IP address of the peer member group.

	Command or Action	Purpose
	Router(config-r-ic)# **member ip 60.60.60.2**	
Step 7	**monitor peer** *bfd* **Example:** Router(config-red)# **monitor peer bfd**	Enables BFD on the POS link.
Step 8	**exit** **Example:** Router(config-r-ic)# **exit**	Exits the redundancy mode.
Step 9	**controller SONET** *slot/bay/port* **Example:** Router(config)# **controller SONET 0/5/2**	Selects and configures a SONET controller and enters controller configuration mode. • *slot*—Chassis slot number, which is always 0. • *bay*—Card interface bay number in a slot. The range is from 0 to 5. • *port*—Port or interface number. The range is from 0 to 7 for Gigabit Ethernet.
Step 10	**framing** [**SDH** | **SONET**] **Example:** Router(config-controller)# **framing SONET**	Configures the controller with framing type. SONET framing is the default option.
Step 11	**clock source internal** **Example:** Router(config-controller)# **clock source internal**	Sets the clocking for individual E1 links.
Step 12	**sts-1** *1-3POS* **Example:** Router(config-controller)# **sts-1 1-3**	Specifies the STS identifier.
Step 13	**exit** **Example:** Router(config-ctrlr-sts1)# **exit**	Exits from the STS configuration mode.
Step 14	**controller SONET** *slot/bay/port* **Example:** Router(config)# **controller SONET 0/5/2**	Selects and configures a SONET controller and enters controller configuration mode.
Step 15	**Shutdown** **Example:** Router(config)# **Shutdown**	Shut down the controller before APS configuration.

	Command or Action	Purpose
Step 16	**aps group** *group_id* **Example:** `Router(config-controller)# aps group 1`	Configures the APS group for POS.
Step 17	**aps protect 1** *remote loopback ip* **Example:** `Router(config-controller)# aps protect 1 192.168.1.1`	Enable the protect node.
Step 18	**aps interchasis group** *interchasis group-id* **Example:** `Router(config-controller)# aps interchasis group 1`	Enable the inter chasis.
Step 19	**no shut** **Example:** `Router(config-controller)# no shut`	Unshut the controller.
Step 20	**exit** **Example:** `Router(config-controller)# exit`	Ends the controller session and returns to the configuration mode.
Step 21	**interface POS** *slot/bay/port* **Example:** `Router(config)# interface POS 0/5/2`	Configures a serial interface and enters the interface configuration mode • *slot*—Chassis slot number, which is always 0. • *bay*—Card interface bay number in a slot. The range is from 0 to 5. • *port*—Port or interface number. The range can be 0-3.
Step 22	**ip address** *ip-address* **Example:** `Router(config-if)# ip address 45.1.1.2 255.255.255.0`	Assigns the ip address to POS interface
Step 23	**encapsulation** *ppp* **Example:** `Router(config-if-srv)# encapsulation ppp`	Specifies the ppp encapsulation over POS interface.
Step 24	**end** **Example:** `Router(config-if-srv)# end`	Returns to privileged EXEC mode.

Verifying MR-APS on POS interface

- Use the **show rgf groups** command to display POS statistics for the configured POS circuits.

  Following is a sample output of the **show rgf groups** command to display the detailed information about POS interface configured on the router:

```
Router# show rgf groups
```

```
Router# sh rgf groups

 Total RGF groups: 2
----------------------------------------------------------
ACTIVE RGF GROUP
 RGF Group ID     : 1
 RGF Peer Group ID: 0
 ICRM Group ID    : 1
 APS Group ID     : 1

RGF State information:
 My State Present  : Active-fast     <<<<<<<<<<Chk this status
         Previous  : Standby-hot
 Peer State Present: Standby-hot
           Previous: Standby-bulk
Misc:
 Communication state Up
 aps_bulk:  0
 aps_stby:  0
 peer_stby: 0
 -> Driven Peer to [Peer Standby Hot] Progression
 -> Standby sent Bulk Sync start Progression
  RGF GET BUF:    66       RGF RET BUF    66
```

  Following is a sample output of the **show ppp interface***POS*

```
Router# show  ppp interface 0/5/2
```

```
PPP Serial Context Info
-------------------
Interface         : PO0/4/2.1
PPP Serial Handle: 0xE9000006
PPP Handle        : 0xBF000006
SSS Handle        : 0x8000006
AAA ID            : 14
Access IE         : 0xA000006
SHDB Handle       : 0xA3000006
State             : Up
Last State        : Binding
Last Event        : LocalTerm
```

- Use the **show ccm group id** *grioup-id number* command to check CCM status

```
Router# show ccm group id
```

```
 CCM Group 1 Details
----------------------------------------

CCM Group ID               : 1
Infra Group ID             : 2
```

```
Infra Type                  : Redundancy Group Facility (RGF)  <<<<Chk this
HA State                    : CCM HA Active
Redundancy State            : Dynamic Sync
Group Initialized/cleaned : FASLE

ASR903_PE2#
```

- Following is a sample output of the **show aps gr 1** command:

```
Router#  show aps gr 1
```

```
SONET 0/4/2 APS Group 1: working channel 1 (Inactive) (HA)
        Protect at 33.1.1.1
        PGP timers (from protect): hello time=1; hold time=10
        SDH framing
        Remote APS configuration: (null)
```

- Following is a sample output of the **show redundancy interchassis** command to display information about interchassis redundancy group configuration:

```
Router# show redundancy interchassis
```

```
Redundancy Group 1 (0x1)
  Applications connected: MSR
  Monitor mode: BFD
  member ip: 10.17.255.163 "ASR903_PE2", CONNECTED
    BFD neighbor: GigabitEthernet0/1/2, next hop 33.1.1.2, DOWN
    MSR state: CONNECTED

ICRM fast-failure detection neighbor table
  IP Address       Status Type Next-hop IP      Interface
  ==========       ====== ==== ===========      =========
  10.17.255.163    DOWN   BFD  33.1.1.2         GigabitEthernet0/1/2
```

Configuration Examples for MR-APS on POS interface

The following example shows how to configure the MR-APS integration on a POS interface on the working router PE1 working node:

```
RouterPE1> enable
RouterPE1(config)#cont so 0/4/2
RouterPE1(config-controller)#au-4 1 pos
RouterPE1(config-controller)#aps gr 1
RouterPE1(config-controller)#aps working 1
RouterPE1(config-controller)#aps interchassis group 1
RouterPE1(config-controller)#exit
RouterPE1(config)#interface POS0/4/2.1
RouterPE1(config-interface)#ip address 45.1.1.2
RouterPE1(config-interface)#encapsulation ppp
RouterPE1(config)# redundancy
RouterPE1(config-red)# interchassis group 1
RouterPE1(config-r-ic)# member ip 14.2.0.2
RouterPE1(config-r-ic)# backbone interface gig 0/0/1
RouterPE1(config-r-ic)# exit
```

The following example shows how to configure the MR-APS integration on a POS interface on the Protect router PE2 Protect node:

```
RouterPE2> enable
RouterPE2(config)#cont so 0/4/2
RouterPE2(config-controller)#framing sdh
RouterPE2(config-controller)#clock source line
RouterPE2(config-controller)#aug mapping au-4
RouterPE2(config-controller)#au-4 1 pos
RouterPE2(config-controller)#aps group 1
RouterPE2(config-controller)#aps protect 1 10.1.1.1
RouterPE2(config-controller)#aps interchassis group 1
RouterPE1(config-controller)#exit
RouterPE2(config)#interface POS0/4/2.1
RouterPE2(config-interface)#ip address 45.1.1.1 255.255.255.0
RouterPE2(config-interface)#encapsulation ppp
RouterPE2(config-controller)#network-clock input-source 1 controller SONET 0/4/2
RouterPE2(config)# redundancy
RouterPE2(config)#mode sso
RouterPE2(config-red)#interchassis group 1
RouterPE2(config-r-ic)#monitor peer bfd
RouterPE2(config-r-ic)#member ip 52.1.1.1
RouterPE2(config-r-ic)# exit
```

The following example shows how to configure the MR-APS integration on a POS interface on the router CE1 working node:

```
RouterPE3> enable
RouterPE3(config)#cont SONET 0/3/1
RouterPE3(config-controller)#framing sdh
RouterPE3(config-controller)#clock source line
RouterPE3(config-controller)#aug mapping au-4
RouterPE3(config-controller)#au-4 1 pos
RouterPE3(config)#interface POS0/4/2.1
RouterPE3(config-interface)#ip address 45.1.1.1
RouterPE3(config-interface)#encapsulation ppp
RouterPE3(config-controller)#network-clock input-source 1 controller SONET 0/4/2
RouterPE1(config-controller)#exit
```

CHAPTER 4

Hot Standby Pseudowire Support for ATM and TDM Access Circuits

Note Hot Standby Pseudowire Support for ATM and IMA circuits are *not* supported on the Cisco ASR 900 RSP3 module.

The Hot Standby Pseudowire Support for ATM and TDM Access Circuits feature is an enhancement to the L2VPN Pseudowire Redundancy feature in the following ways:

- Faster failover of to the backup pseudowire
- Less traffic loss during failover

The Hot Standby Pseudowire Support for ATM and TDM Access Circuits feature allows the backup pseudowire to be in a "hot standby" state, so that it can immediately take over if the primary pseudowire fails. The following sections explain the concepts and configuration tasks for this feature.

Prerequisites for Hot Standby Pseudowire Support for ATM and TDM Access Circuits

- This feature requires that you understand how to configure Layer 2 virtual private networks (VPNs). You can find that information in the following documents:
  - Any Transport over MPLS
  - L2 VPN Interworking
  - L2VPN Pseudowire Redundancy

- The Hot Standby Pseudowire Support for ATM and TDM Access Circuits feature recommends that the following mechanisms be in place to enable faster detection of a failure in the network:
  - Label-switched paths (LSP) Ping/Traceroute and Any Transport over MPLS Virtual Circuit Connection Verification (AToM VCCV)
  - Local Management Interface (LMI)
  - Operation, Administration, and Maintenance (OAM)

Restrictions for Hot Standby Pseudowire Support for ATM and TDM Access Circuits

- Hot Standby Pseudowire Support for ATM and TDM Access Circuits is *not* supported on L2TPv3. Only MPLS L2VPNs are supported.
- Hot Standby Pseudowire Support for ATM and IMA is *not* supported on the Cisco ASR 900 RSP3 module.
- More than one backup pseudowire is *not* supported.
- Different pseudowire encapsulation types on the MPLS pseudowire are not supported.
- If you use Hot Standby Pseudowire Support for ATM and TDM Access Circuits with L2VPN Interworking, the interworking method must be the same for the primary and backup pseudowires. For TDM access circuits, interworking is *not* supported.
- Only dynamic pseudowires are supported.
- Pseudowire over static VPLS is *not* supported on the Cisco ASR 900 RSP3 module.

Information About Hot Standby Pseudowire Support for ATM and TDM Access Circuits

How the Hot Standby Pseudowire Support for ATM and TDM Access Circuits Feature Works

The Hot Standby Pseudowire Support for ATM and TDM Access Circuits feature improves the availability of L2VPN pseudowires by detecting failures and handling them with minimal disruption to the service.

The Hot Standby Pseudowire Support for ATM and TDM Access Circuits feature allows the backup pseudowire to be in a “hot standby” state, so that it can immediately take over if the primary pseudowire fails. The L2VPN Pseudowire Redundancy feature allows you to configure a backup pseudowire too, but in a cold state. With the L2VPN Pseudowire Redundancy feature, if the primary pseudowire fails, it takes time for the backup pseudowire to take over, which causes a loss in traffic.

If you have configured L2VPN Pseudowire Redundancy on your network and upgrade to Cisco IOS Release 15.1(1)S, you do not need add any other commands to achieve Hot Standby Pseudowire Support for ATM and TDM Access Circuits. The backup pseudowire will automatically be in a hot standby state.

Supported Transport Types

The Hot Standby Pseudowire Support for ATM and TDM Access Circuits feature supports the following transport types:

- ATM
    - ATM AAL5 in VC mode
    - ATM packed cell relay in VC Mode
    - ATM in VP mode
    - ATM packed cell relay in VP mode
    - ATM in port mode
    - ATM packed cell relay in port mode
- Time division multiplexing (TDM)
    - Structure-Agnostic TDM over Packet (SAToP)
    - Circuit Emulation Services over PSN (CESoPSN)

How to Configure Hot Standby Pseudowire Support for ATM and TDM Access Circuits

The Hot Standby Pseudowire Support for ATM and TDM Access Circuits feature enables you to configure a backup pseudowire in case the primary pseudowire fails. When the primary pseudowire fails, the PE router can immediately switch to the backup pseudowire.

Configuring a Pseudowire for Static VPLS

Note Pseudowire for Static VPLS is *not* supported on the Cisco ASR 900 RSP3 module.

The configuration of pseudowires between provider edge (PE) devices helps in the successful transmission of the Layer 2 frames between PE devices.

Use the pseudowire template to configure the virtual circuit (VC) type for the virtual path identifier (VPI) pseudowire. In the following task, the pseudowire will go through a Multiprotocol Label Switching (MPLS)-Tunneling Protocol (TP) tunnel.

The pseudowire template configuration specifies the characteristics of the tunneling mechanism that is used by the pseudowires, which are:

- Encapsulation type
- Control protocol
- Payload-specific options
- Preferred path

Perform this task to configure a pseudowire template for static Virtual Private LAN Services (VPLS).

Note Ensure that you perform this task before configuring the virtual forwarding instance (VFI) peer. If the VFI peer is configured before the pseudowire class, the configuration is incomplete until the pseudowire class is configured. The **show running-config** command displays an error stating that configuration is incomplete.

```
Device# show running-config | sec vfi

l2 vfi config manual
 vpn id 1000
 ! Incomplete point-to-multipoint vfi config
```

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**template type pseudowire** *name* **Example:** `Device(config)# template type pseudowire static-vpls`	Specifies the template type as pseudowire and enters template configuration mode.
Step 4	**encapsulation mpls** **Example:** `Device(config-template)# encapsulation mpls`	Specifies the tunneling encapsulation. • For Any Transport over MPLS (AToM), the encapsulation type is MPLS.
Step 5	**signaling protocol none** **Example:** `Device(config-template)# signaling protocol none`	Specifies that no signaling protocol is configured for the pseudowire class.
Step 6	**preferred-path interface Tunnel-tp** *interface-number* **Example:** `Device(config-template)# preferred-path interface Tunnel-tp 1`	(Optional) Specifies the path that traffic uses: an MPLS Traffic Engineering (TE) tunnel or destination IP address and Domain Name Server (DNS) name.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(config-template)# exit`	Exits template configuration mode and returns to global configuration mode.
Step 8	**interface pseudowire** *number* **Example:** `Device(config)# interface pseudowire 1`	Establishes a pseudowire interface and enters interface configuration mode.
Step 9	**source template type pseudowire** *name* **Example:** `Device(config-if)# source template type pseudowire static-vpls`	Configures the source template type of the configured pseudowire.
Step 10	**neighbor** *peer-address vcid-value* **Example:** `Device(config-if)# neighbor 10.0.0.1 123`	Specifies the peer IP address and VC ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 11	**label** *local-pseudowire-label remote-pseudowire-label* **Example:** `Device(config-if)# label 301 17`	Configures an Any Transport over MPLS (AToM) static pseudowire connection by defining local and remote circuit labels.
Step 12	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Hot Standby Pseudowire Support for ATM and TDM Access Circuits

Use the following steps to configure the Hot Standby Pseudowire Support for ATM and TDM Access Circuits feature.

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *number* **Example:** `Router(config)# interface atm4/1/0`	Specifies the ATM interface and enters interface configuration mode.
Step 4	**pvc** [*name*] *vpi/vci* **l2transport** **Example:** `Router(config-if)# pvc 1/100 l2transport`	Creates or assigns a name to an ATM PVC and enters L2transport PVC configuration mode.
Step 5	**xconnect** *peer-router-id vcid* {**encapsulation mpls**| **pw-class** *pw-class-name}* **Example:** `Router(config-if-atm-l2trans-pvc)# xconnect 10.0.0.1 123 pw-class atom`	Binds the attachment circuit to a pseudowire VC.
Step 6	**backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*] **Example:** `Router(config-if-atm-l2trans-pvc)# backup peer 10.0.0.3 125 pw-class atom`	Specifies a redundant peer for the pseudowire VC. The pseudowire class name must match the name you specified when you created the pseudowire class, but you can use a different pw-class in the **backup peer** command than the name that you used in the primary **xconnect** command.
Step 7	**backup delay** *enable-delay* {*disable-delay* | **never**} **Example:** `Router(config-if-atm-l2trans-pvc)# backup delay 5 never`	Specifies how long (in seconds) the backup pseudowire VC should wait to take over after the primary pseudowire VC goes down. The range is 0 to 180. Specifies how long the primary pseudowire should wait after it becomes active to take over for the backup pseudowire VC. The range is 0 to 180 seconds. If you specify the **never keyword**, the primary pseudowire VC never takes over for the backup.

Verifying the Hot Standby Pseudowire Support for ATM and TDM Access Circuits Configuration

Use the following commands to verify that the backup pseudowire is provisioned for hot standby support.

Procedure

Step 1 **show atm acircuit**

If the output of the **show atm acircuit**command shows two entries for the same vpi/vci, then the backup pseudowire has been correctly provisioned, as shown in the following example:

Example:

```
Router# show atm acircuit

Interface          VPI  VCI    AC    Id       Switch   Segment  St  Flg  Prov
---------          ---  ---    --    --       -------  -------- --  ---  ----
ATM2/1/0.2         11   111    ATA5  1         2003    4007     2    0    Y
ATM2/1/0.2         11   111    ATA5  1         1002    3006     2    0    Y
```

Step 2 **show atm pvc**

If the output of the **show atm pvc command includes "Red Prov: Yes," then the backup p**seudowire has been correctly provisioned, as shown in bold in the following example:

Example:

```
Router# show atm pvc 1/1010
Interworking Method: like to like
AC Type: ATM AAL5,  Circuit Id: 2,  AC State: UP,  Prov: YES
Switch Hdl: 0x1005,  Segment hdl: 0x4011
Red Switch Hdl: 0x3007, Red Segment hdl: 0x6010, Red Prov: YES
AC Hdl: 0x7200000F,  AC Peer Hdl: 0x5D000012,  Flg:0,  Platform Idx:10
Status: UP
```

Step 3 **show cem acircuit**

If the output of the **show cem acircuit command includes "Redundancy Member Prov: Yes," then the backup p**seudowire has been correctly provisioned, as shown in bold in the following example:

Example:

```
Router# show cem acircuit
CEM Int.   ID  Flags  Swhdl    Seghdl    Ckttype  Provisioned
----------------------------------------------------------------
CEM3/0/0   1    0    B00E       201E       19        Yes
Redundancy Switch hdl: 0xC00F Redundancy Segment hdl: 0x401F   Redundancy Member Prov: Yes
```

Step 4 **show cem acircuit detail**

If the output of the **show cem acircuit detail command includes "Redundancy Member Prov: Yes," then the backup p**seudowire has been correctly provisioned, as shown in bold in the following example:

Example:

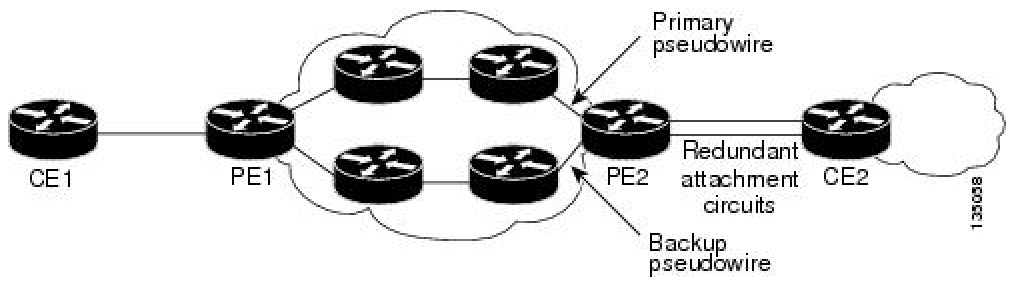
```
Router# show cem acircuit detail

CEM3/0/0    Cemid 1
PW Ckt_type: 19  Aie hdl: EE00000B Peer aie hdl: 0x2000000C
Switch hdl: 0xB00E     Segment hdl: 0x201E     Redundancy Switch hdl: 0x1000     Redundancy
 Segment hdl: 0x4002     Redundancy Member Prov: Yes
```

Configuration Examples for Hot Standby Pseudowire Support for ATM and TDM Access Circuits

Configuring Hot Standby Pseudowire Support for ATM and TDM Access Circuits on CEM Circuits Example

The figure below shows the configuration of Hot Standby Pseudowire Support for ATM and TDM Access Circuits, where the backup pseudowire is on the same PE router.

Figure 9: Hot Standby Pseudowire Topology



The configuration shown in the figure above is used in the following examples:

Table 4: Configuring Hot Standby Pseudowire Support for ATM and TDM Access Circuits on CEM Circuits: Example

[illegible]

Table 5: Configuring Hot Standby Pseudowire Support for ATM and TDM Access Circuits on ATM Circuits: Example

PE1	PE2

`interface Loopback0`
` ip address 10.44.44.44 255.255.255.255`
`!`
`interface POS3/3/0`
` ip address 10.4.4.4 255.255.255.0`
` mpls ip`
`!`
`interface ATM4/1/0`
` no ip address`
` no atm enable-ilmi-trap`
` pvc 1/100 l2transport`
`  xconnect 10.22.22.22 1 encapsulation mpls`
 	
`   backup peer 10.22.22.22 2`
	`interface Loopback0`
	` ip address 10.22.22.22 255.255.255.255`
	`!`
	`interface POS3/3/0`
	` ip address 10.4.4.1 255.255.255.0`
	` mpls ip`
	`!`
	`interface ATM4/1/0`
	` no ip address`
	` no atm enable-ilmi-trap`
	` pvc 1/100 l2transport`
	`  xconnect 10.44.44.44 1 encapsulation mpls`
	`!`
	` pvc 1/200 l2transport`
	`  xconnect 10.44.44.44 2 encapsulation mpls`

CHAPTER 5

PPP and Multilink PPP Configuration

Note PPP and Multilink PPP Configuration is *not* supported on the Cisco ASR 900 RSP3 module.

This module describes how to configure PPP and Multilink PPP (MLP) features on any interface. Multilink PPP provides a method for spreading traffic across multiple physical WAN links.

Limitations

The following limitations apply when using MLPPP on the Cisco ASR 903 Router:

- All links in an MLPPP bundle must be on the same interface module.
- All links in an MLPPP bundle must be of the same bandwidth.
- The router supports a maximum of 16 links per bundle and a minimum of 2 links per bundle. Maximum number of bundles supported per interface module is 168.
- To change the MLPPP bundle fragmentation mode between enabled and disabled, perform a **shutdown**/**no shutdown** on the bundle.
- LFI is not supported. However, PPP Multilink fragmentation is supported by default. To disable fragmentation, see Disabling PPP Multilink Fragmentation.
- Multicast MLP is not supported.
- PPP compression is not supported.
- PPP half bridging is not supported.
- IPv6 is not supported for this feature.
- To enable an ACFC or PFC configuration, issue a shut **shutdown**/**no shutdown** on the serial interface.

- Channelization is not supported
- Also that only 1 channel-group can be created per controller with complete timeslots.
- PPP and MLPPP are supported on synchronous serial interfaces; Asynchronous serial interfaces, high-speed serial interfaces (HSSI), and ISDN interfaces are not supported.
- If you configure interfaces on each end of an MLPPP connection with different MTU values, the link drops traffic at high traffic rates. We recommend that you configure the same MTU values across all nodes in an MLPPP connection.

PPP and Multilink PPP

To configure the Media-Independent PPP and Multilink PPP, you should understand the following concepts:

Point-to-Point Protocol

Point-to-Point Protocol (PPP), described in RFC 1661, encapsulates network layer protocol information over point-to-point links. You can configure PPP on synchronous serial interfaces.

Challenge Handshake Authentication Protocol (CHAP), Microsoft Challenge Handshake Authentication Protocol (MS-CHAP), or Password Authentication Protocol (PAP)

Magic Number support is available on all serial interfaces. PPP always attempts to negotiate for Magic Numbers, which are used to detect looped-back lines. Depending on how the **down-when-looped** command is configured, the router might shut down a link if it detects a loop.

CHAP or PPP Authentication

PPP with CHAP or PAP authentication is often used to inform the central site about which remote routers are connected to it.

With this authentication information, if the router or access server receives another packet for a destination to which it is already connected, it does not place an additional call. However, if the router or access server is using rotaries, it sends the packet out the correct port.

CHAP and PAP were originally specified in RFC 1334, and CHAP was updated in RFC 1994. These protocols are supported on synchronous and asynchronous serial interfaces. When using CHAP or PAP authentication, each router or access server identifies itself by a name. This identification process prevents a router from placing another call to a router to which it is already connected, and also prevents unauthorized access.

Access control using CHAP or PAP is available on all serial interfaces that use PPP encapsulation. The authentication feature reduces the risk of security violations on your router or access server. You can configure either CHAP or PAP for the interface.

Note To use CHAP or PAP, you must be running PPP encapsulation.

When CHAP is enabled on an interface and a remote device attempts to connect to it, the local router or access server sends a CHAP packet to the remote device. The CHAP packet requests or “challenges” the remote device to respond. The challenge packet consists of an ID, a random number, and the hostname of the local router.

The required response has two parts:

- An encrypted version of the ID, a secret password, and the random number
- Either the hostname of the remote device or the name of the user on the remote device

When the local router or access server receives the response, it verifies the secret password by performing the same encryption operation as indicated in the response and looking up the required hostname or username. The secret passwords must be identical on the remote device and the local router.

Because this response is sent, the password is never sent in clear text, preventing other devices from stealing it and gaining illegal access to the system. Without the proper response, the remote device cannot connect to the local router.

CHAP transactions occur only when a link is established. The local router or access server does not request a password during the rest of the call. (The local device can, however, respond to such requests from other devices during a call.)

When PAP is enabled, the remote router attempting to connect to the local router or access server is required to send an authentication request. The username and password specified in the authentication request are accepted, and the Cisco IOS software sends an authentication acknowledgment.

After you have enabled CHAP or PAP, the local router or access server requires authentication from remote devices. If the remote device does not support the enabled protocol, no traffic will be passed to that device.

To use CHAP or PAP, you must perform the following tasks:

- Enable PPP encapsulation.
- Enable CHAP or PAP on the interface.

For CHAP, configure hostname authentication and the secret password for each remote system with which authentication is required.

IP Address Pooling

A point-to-point interface must be able to provide a remote node with its IP address through the IP Control Protocol (IPCP) address negotiation process. The IP address can be obtained from a variety of sources. The address can be configured through the command line, entered with an EXEC-level command, provided by TACACS+ or the Dynamic Host Configuration Protocol (DHCP), or from a locally administered pool.

IP address pooling uses a pool of IP addresses from which an incoming interface can provide an IP address to a remote node through IPCP address negotiation process. IP address pooling also enhances configuration flexibility by allowing multiple types of pooling to be active simultaneously.

The IP address pooling feature allows configuration of a global default address pooling mechanism, per-interface configuration of the address pooling mechanism, and per-interface configuration of a specific address or pool name.

Peer Address Allocation

A peer IP address can be allocated to an interface through several methods:

- Dialer map lookup—This method is used only if the peer requests an IP address, no other peer IP address has been assigned, and the interface is a member of a dialer group.
- PPP EXEC command—An asynchronous dialup user can enter a peer IP address or hostname when PPP is invoked from the command line. The address is used for the current session and then discarded.
- IPCP negotiation—If the peer presents a peer IP address during IPCP address negotiation and no other peer address is assigned, the presented address is acknowledged and used in the current session.
- Default IP address.
- TACACS+ assigned IP address—During the authorization phase of IPCP address negotiation, TACACS+ can return an IP address that the user being authenticated on a dialup interface can use. This address overrides any default IP address and prevents pooling from taking place.
- DHCP retrieved IP address—If configured, the routers acts as a proxy client for the dialup user and retrieves an IP address from a DHCP server. That address is returned to the DHCP server when the timer expires or when the interface goes down.
- Local address pool—The local address pool contains a set of contiguous IP addresses (a maximum of 1024 addresses) stored in two queues. The free queue contains addresses available to be assigned and the used queue contains addresses that are in use. Addresses are stored to the free queue in first-in, first-out (FIFO) order to minimize the chance the address will be reused, and to allow a peer to reconnect using the same address that it used in the last connection. If the address is available, it is assigned; if not, another address from the free queue is assigned.
- Chat script (asynchronous serial interfaces only)—The IP address in the **dialer map** command entry that started the script is assigned to the interface and overrides any previously assigned peer IP address.
- Virtual terminal/protocol translation—The translate command can define the peer IP address for a virtual terminal (pseudo asynchronous interface).
- The pool configured for the interface is used, unless TACACS+ returns a pool name as part of authentication, authorization, and accounting (AAA). If no pool is associated with a given interface, the global pool named default is used.

Precedence Rules

The following precedence rules of peer IP address support determine which address is used. Precedence is listed from most likely to least likely:

1. AAA/TACACS+ provided address or addresses from the pool named by AAA/TACACS+
2. An address from a local IP address pool or DHCP (typically not allocated unless no other address exists)
3. Dialer map lookup address (not done unless no other address exists)
4. Address from an EXEC-level PPP command, or from a chat script
5. Configured address from the **peer default ip address** command or address from the protocol **translate** command
6. Peer-provided address from IPCP negotiation (not accepted unless no other address exists)

MLP on Synchronous Serial Interfaces

Address pooling is available on all synchronous serial interfaces that are running PPP and PPPoX sessions.

MLP provides characteristics are most similar to hardware inverse multiplexers, with good manageability and Layer 3 services support. Figure below shows a typical inverse multiplexing application using two Cisco routers and Multilink PPP over four T1 lines.

How to Configure PPP

The sections below describe how to configure PPP.

Enabling PPP Encapsulation

The **encapsulation ppp** command enables PPP on serial lines to encapsulate IP and other network protocol datagrams.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *slot/subslot/port:channel* **Example:** `Router(config)# interface serial 0/0/0:0`	Enters interface configuration mode.
Step 4	**encapsulation ppp** **Example:** `Router(config-if) # encapsulation ppp`	Enables PPP encapsulation. **Note** PPP echo requests are used as keepalives to minimize disruptions to the end users of your network. Use the **no keepalive** command to disable echo requests.
Step 5	**end** **Example:**	Exits interface configuration mode.

	Command or Action	Purpose
	`Router(config-if)# end`	

Enabling CHAP or PAP Authentication

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *number* **Example:** `Router(config)# interface serial 0/0/0`	Enters Interface Configuration mode.
Step 4	**ppp authentication** {**chap** | **chap pap** | **pap chap** | *pap*} [**if-needed**] [*list-name* | **default**] [**callin**] **Example:** `Router(config-if)# ppp authentication chap`	Defines the authentication methods supported and the order in which they are used. **Note** • Use the **ppp authentication chap** command only with TACACS or extended TACACS. • With AAA configured on the router and list names defined for AAA, the *list-name* optional argument can be used with AAA/TACACS+. Use the **ppp use-tacacs** command with TACACS and Extended TACACS. Use the **aaa authentication ppp** command with AAA/TACACS+.
Step 5	**ppp use-tacacs** [**single-line**]or **aaa authentication ppp** **Example:**	Configure TACACS on a specific interface as an alternative to global host authentication.

	Command or Action	Purpose
	`Router(config-if)# ppp use-tacacs single-line` `Router(config-if)# aaa authentication ppp`	
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 7	**username** *name* [**user-maxlinks** *link-number*] **password** *secret* **Example:** `Router(config)# username name user-maxlinks 1 password password1`	Configures identification. • Optionally, you can specify the maximum number of connections a user can establish. • To use the **user-maxlinks** keyword, you must also use the **aaa authorization network default local** command and PPP encapsulation and name authentication on all the interfaces the user will be accessing.
Step 8	**end** **Example:** `Router(config)# end`	Exits global configuration mode and enters privileged EXEC mode. **Caution** If you use a list name that has not been configured with the **aaa authentication ppp** command, you disable PPP on the line.

Example

```
Router# configure terminal
Router(config)# interface serial 0/0/0
Router(config-if)# ppp authentication chap
Router(config-if)# aaa authentication ppp
Router(config-if)# exit
Router(config)# username name user-maxlinks 1 password password1
Router(config)# end
```

Configuring IP Address Pooling

You can define the type of IP address pooling mechanism used on router interfaces in one or both of the ways described in the following sections:

Note For more information about address pooling, see the IP Addressing Configuration Guide Library, Cisco IOS XE Release 3S

Global Default Address Pooling Mechanism

The global default mechanism applies to all point-to-point interfaces that support PPP encapsulation and that have not otherwise been configured for IP address pooling. You can define the global default mechanism to be either DHCP or local address pooling.

To configure the global default mechanism for IP address pooling, perform the tasks in the following sections:

- Defining DHCP as the Global Default Mechanism
- Defining Local Address Pooling as the Global Default Mechanism

After you have defined a global default mechanism, you can disable it on a specific interface by configuring the interface for some other pooling mechanism. You can define a local pool other than the default pool for the interface or you can configure the interface with a specific IP address to be used for dial-in peers.

You can also control the DHCP network discovery mechanism; see the following section for more information:

- Controlling DHCP Network Discovery

Defining DHCP as the Global Default Mechanism

DHCP specifies the following components:

- A DHCP server—A host-based DHCP server configured to accept and process requests for temporary IP addresses.
- A DHCP proxy client—A Cisco access server configured to arbitrate DHCP calls between the DHCP server and the DHCP client. The DHCP client-proxy feature manages a pool of IP addresses available to dial-in clients without a known IP address.

Perform this task to enable DHCP as the global default mechanism.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip address-pool dhcp-proxy-client** **Example:** `Router(config)# ip address-pool dhcp-proxy-client`	Specifies the DHCP client-proxy feature as the global default mechanism. • The **peer default ip address** command and the **member peer default ip address** command can be used to define default peer IP addresses.

	Command or Action	Purpose
		Note You can provide as few as one or as many as ten DHCP servers for the proxy client (the Cisco router or access server) to use. The DHCP servers provide temporary IP addresses.
Step 4	**ip dhcp-server** [*ip-address* | *name*] **Example:** `Router(config)# ip dhcp-server 209.165.201.1`	(Optional) Specifies the IP address of a DHCP server for the proxy client to use.
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

Defining Local Address Pooling as the Global Default Mechanism

Perform this task to define local address pooling as the global default mechanism.

Note If no other pool is defined, a local pool called "default" is used. Optionally, you can associate an address pool with a named pool group.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip address-pool local** **Example:** `Router(config)# ip address-pool local`	Specifies local address pooling as the global default mechanism.

	Command or Action	Purpose
Step 4	**ip local pool** {*named-address-pool* | **default**} *first-IP-address* [*last-IP-address*] [**group** *group-name*] [**cache-size** *size*] **Example:** `Router(config)# ip local pool default 192.0.2.1`	Creates one or more local IP address pools.

Controlling DHCP Network Discovery

Perform the steps in this section to allow peer routers to dynamically discover Domain Name System (DNS) and NetBIOS name server information configured on a DHCP server using PPP IPCP extensions.

The **ip dhcp-client network-discovery** global configuration command provides a way to control the DHCP network discovery mechanism. The number of DHCP Inform or Discovery messages can be set to 1 or 2, which determines how many times the system sends the DHCP Inform or Discover messages before stopping network discovery. You can set a timeout period from 3 to 15 seconds, or leave the default timeout period at 15 seconds. The default for the **informs** and **discovers** keywords is 0, which disables the transmission of these messages.

Note For more information about DHCP, see the IP Addressing Configuration Guide Library, Cisco IOS XE Release 3S

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp-client network-discovery** *informs number-of-messages* **discovers** *number-of-messages* **period** *seconds* **Example:** `Router(config)# ip dhcp-client network-discovery informs 2 discovers 2 period 2`	Provides control of the DHCP network discovery mechanism by allowing the number of DHCP Inform and Discover messages to be sent, and a timeout period for retransmission, to be configured.

Configuring IP Address Assignment

Perform this task to configure IP address alignment.

After you have defined a global default mechanism for assigning IP addresses to dial-in peers, you can configure the few interfaces for which it is important to have a nondefault configuration. You can do any of the following;

- Define a nondefault address pool for use by a specific interface.
- Define DHCP on an interface even if you have defined local pooling as the global default mechanism.
- Specify one IP address to be assigned to all dial-in peers on an interface.
- Make temporary IP addresses available on a per-interface basis to asynchronous clients using PPP.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip local pool** {*named-address-pool* | **default**} {*first-IP-address* [*last-IP-address*]} [**group** *group-name*] [**cache-size** *size*]} **Example:** `Router(config)# ip local pool default 192.0.2.0`	Creates one or more local IP address pools.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 2/0`	Specifies the interface and enters interface configuration mode.
Step 5	**peer default ip address pool** *pool-name-list* **Example:** `Router(config-if)# peer default ip address pool 2`	Specifies the pool or pools for the interface to use.
Step 6	**peer default ip address pool dhcp** **Example:**	Specifies DHCP as the IP address mechanism on this interface.

	Command or Action	Purpose
	`Router(config-if)# peer default ip address pool dhcp`	
Step 7	**peer default ip address** *ip-address* **Example:** `Router(config-if)# peer default ip address 192.0.2.2`	Specifies the IP address to assign to all dial-in peers on an interface.

Disabling or Reenabling Peer Neighbor Routes

The Cisco IOS software automatically creates neighbor routes by default; that is, it automatically sets up a route to the peer address on a point-to-point interface when the PPP IPCP negotiation is completed.

To disable this default behavior or to reenable it once it has been disabled, perform the following task:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal` Enters global configuration mode.	
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/1`	Specifies the interface and enters interface configuration mode.
Step 4	**no peer neighbor-route** **Example:** `Router(config-if)# no peer neighbor-route`	Disables creation of neighbor routes.
Step 5	**peer neighbor-route** **Example:** `Router(config-if)# peer neighbor-route`	Reenables creation of neighbor routes. **Note** If entered on a dialer or asynchronous group interface, this command affects all member interfaces.

Configuring Multilink PPP

The Multilink PPP feature provides load balancing functionality over multiple WAN links, while providing multivendor interoperability, packet fragmentation and proper sequencing, and load calculation on both inbound and outbound traffic. The Cisco implementation of MLP supports the fragmentation and packet sequencing specifications in RFC 1990. Additionally, you can change the default endpoint discriminator value that is supplied as part of user authentication. Refer to RFC 1990 for more information about the endpoint discriminator.

MLP allows packets to be fragmented and the fragments to be sent at the same time over multiple point-to-point links to the same remote address. The multiple links come up in response to a defined dialer load threshold. The load can be calculated on inbound traffic, outbound traffic, or on either, as needed for the traffic between the specific sites. MLP provides bandwidth on demand and reduces transmission latency across WAN links.

MLP is designed to work over synchronous and asynchronous serial and BRI and PRI types of single or multiple interfaces that have been configured to support both dial-on-demand rotary groups and PPP encapsulation.

Perform the tasks in the following sections, as required for your network, to configure MLP:

Configuring MLP on Synchronous Interfaces

To configure Multilink PPP on synchronous interfaces, you configure the synchronous interfaces to support PPP encapsulation and Multilink PPP.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *number* **Example:** `Router(config)# interface serial 0/0/1`	Specifies an asynchronous interface and enters interface configuration mode.
Step 4	**no ip address** **Example:** `Router(config-if)# no ip address`	Specifies no IP address for the interface.
Step 5	**encapsulation ppp** **Example:**	Enables PPP encapsulation.

	Command or Action	Purpose
	`Router(config-if)# encapsulation ppp`	
Step 6	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables Multilink PPP.
Step 7	**pulse-time** *seconds* **Example:** `Router(config-if)#  pulse-time 60`	Enables pulsing data terminal ready (DTR) signal intervals on an interface. **Note** Repeat these steps for additional synchronous interfaces, as needed.

Configuring a Multilink Group

A multilink group allows you to assign multiple interfaces to a multilink bundle. When the **ppp multilink group** command is configured on an interface, the interface is restricted from joining any interface but the designated multilink group interface. If a peer at the other end of the interface tries to join a different multilink group, the connection is severed. This restriction applies when Multilink PPP (MLP) is negotiated between the local end and the peer system. The interface can still come up as a regular PPP interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Router(config)# interface multilink 2`	Creates a multilink bundle and enters interface configuration mode to configure the bundle.
Step 4	**ip address** *address mask* **Example:** `Router(config-if)# ip address 192.0.2.1 255.255.255.224`	Sets a primary IP address for an interface.
Step 5	**encapsulation ppp** **Example:**	Enables PPP encapsulation.

	Command or Action	Purpose
	`Router(config-if)# encapsulation ppp`	
Step 6	**ppp chap hostname** *hostname* **Example:** `Router(config-if)# ppp chap hostname host1`	Specifies the hostname on the interface.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 8	**interface** *type number* **Example:** `Router(config)# interface serial 0/0/1`	Enters interface configuration mode.
Step 9	**ppp multilink group** *group-number* **Example:** `Router(config-if)# ppp multilink group 2`	Restricts a physical link to joining only a designated multilink group interface.
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface 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.

Configuring ACFC

Follow these steps to configure ACFC handling during PPP negotiation

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	

	Command or Action	Purpose
	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(config)# interface multilink 2`	Select a multilink interface.
Step 4	**ppp acfc local** {**request** | **forbid**} **Example:** `Router(config-if)# ppp acfc local request`	Configure how the router handles ACFC in its outbound configuration requests 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.
Step 5	**ppp acfc remote** {**apply** | **reject** | **ignore**} **Example:** `Router(config-if)# ppp acfc remote apply`	Configure how the router handles the ACFC option in configuration requests received from a remote peer 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.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.

Configuring PFC

Follow these steps to configure PFC handling during PPP negotiation:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable` Enables privileged EXEC mode. • Enter your password if prompted.	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *number* **Example:** `Router(config)# interface multilink 2`	Select a multilink interface.
Step 4	**ppp pfc local {request | forbid**}`Router(config-if)# ppp pfc local request`	Configure how the router handles PFC in its outbound configuration requests 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.
Step 5	**ppp pfc remote** {**apply** | **reject** | **ignore**} **Example:** `Router(config-if)# ppp pfc remote apply`	Configure a method for the router to use to manage the PFC option in configuration requests received from a remote peer where: • **apply**—PFC options are accepted and PFC may be performed on frames sent to the remote peer. • **reject**—PFC options are explicitly ignored. • **ignore**—PFC options are accepted, but PFC is not performed on frames sent to the remote peer.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.

Changing the Default Endpoint Discriminator

By default, when the system negotiates use of MLP with the peer, the value that is supplied for the endpoint discriminator is the same as the username used for authentication. That username is configured for the interface by the Cisco IOS **ppp chap hostname** or **ppp pap sent-username** command, or defaults to the globally configured hostname (or stack group name, if this interface is a Stack Group Bidding Protocol, or SGBP, group member).

Perform this task to override or change the default endpoint discriminator.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual template** *number* **Example:** `Router(config)# interface virtual template 1`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces and enters interface configuration mode.
Step 4	**ppp multilink endpoint** {**hostname** | **ip** *ipaddress* | **mac** *LAN-interface* | **none** | **phone** *telephone-number* | **string** *char-string*} **Example:** `Router(config-if)# ppp multilink endpoint ip 192.0.2.0`	Overrides or changes the default endpoint discriminator the system uses when negotiating the use of MLP with the peer.

Creating a Multilink Bundle

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Router(config)# interface multilink 10`	Assigns a multilink group number and enters interface configuration mode.
Step 4	**ip address** *address mask* **Example:** `Router(config-if)# ip address 192.0.2.9 255.255.255.224`	Assigns an IP address to the multilink interface.
Step 5	**encapsulation ppp** **Example:** `Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 6	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables Multilink PPP.

Assigning an Interface to a Multilink Bundle

Caution Do not install a router to the peer address while configuring an MLP lease line. This installation can be disabled when **no ppp peer-neighbor-route** command is used under the MLPPP bundle interface.

Procedure

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

	Command or Action	Purpose
Step 3	**interface multilink** *group-number* **Example:** `Router(config)# interface multilink 10`	Assigns a multilink group number and enters interface configuration mode.
Step 4	**no ip address** **Example:** `Router(config-if)# no ip address`	Removes any specified IP address.
Step 5	**keepalive** **Example:** `Router(config-if)# keepalive`	Sets the frequency of keepalive packets.
Step 6	**encapsulation ppp** **Example:** `Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 7	**ppp multilink group** *group-number* **Example:** `Router(config-if)# ppp multilink 12`	Restricts a physical link to joining only the designated multilink-group interface.
Step 8	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables Multilink PPP.
Step 9	**ppp authentication chap** **Example:** `Router(config-if)# ppp authentication chap`	(Optional) Enables CHAP authentication.
Step 10	**pulse-time** *seconds* **Example:** `Router(config-if)# pulse-time 10`	(Optional) Configures DTR signal pulsing.

Configuring PPP/MLP MRRU Negotiation Configuration on Multilink Groups

In this task, you configure MRRU negotiation on the multilink interface. The bundle interface is static, that is, always available.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *number* **Example:** `Router(config)# interface multilink 10`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.13.1.1 255.255.255.0`	Sets the IP address for the interface.
Step 5	**ppp multilink mrru** [**local** | **remote**] *mrru-value* **Example:** `Router(config-if)# ppp multilink mrru local 1600`	Configures the MRRU value negotiated on a multilink bundle when MLP is used. • **local**—(Optional) Configures the local MRRU value. The default values for the local MRRU are the value of the multilink group interface MTU for multilink group members, and 1524 bytes for all other interfaces. • **remote**—(Optional) Configures the minimum value that the software will accept from the peer when it advertises its MRRU. By default, the software accepts any peer MRRU value of 128 or higher. You can specify a higher minimum acceptable MRRU value in a range from 128 to 16384 bytes.
Step 6	**mtu** bytes **Example:** `Router(config-if)# mtu 1600`	(Optional) Adjusts the maximum packet size or MTU size. • Once you configure the MRRU on the bundle interface, you enable the router to receive large reconstructed MLP frames. You may want to configure the bundle MTU so the router can transmit large

	Command or Action	Purpose
		MLP frames, although it is not strictly necessary. • The maximum recommended value for the bundle MTU is the value of the peer's MRRU. The default MTU for serial interfaces is 1500. The software will automatically reduce the bundle interface MTU if necessary, to avoid violating the peer's MRRU.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface serial** *slot/port* **Example:** `Router(config)# interface serial 0/0`	Selects a serial interface to configure and enters interface configuration mode.
Step 9	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables MLP on the interface.
Step 10	**ppp multilink group** *group-number* **Example:** `Router(config-if)# ppp multilink group 1`	Restricts a physical link to joining only a designated multilink-group interface.
Step 11	**mtu** *bytes* **Example:** `Router(config-if)# mtu 1600`	(Optional) Adjusts the maximum packet size or MTU size. • The default MTU for serial interfaces is 1500. • When the bundle interface MTU is tuned to a higher number, then depending upon the fragmentation configuration, the link interface may be given larger frames to transmit. • You must ensure that fragmentation is performed such that fragments are sized less than the link interface MTU (refer to command pages for the **ppp multilink fragmentation** and **ppp multilink fragment-delay** commands for more

	Command or Action	Purpose
		information about packet fragments), or configure the MTUs of the link interfaces such that they can transmit the larger frames.
Step 12	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Disabling PPP Multilink Fragmentation

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Router(config)# interface multilink 10`	Assigns a multilink group number and enters interface configuration mode.
Step 4	**ppp multilink fragment disable** **Example:** `Router(config-if)# ppp multilink fragment disable`	(Optional) Disables PPP multilink fragmentation.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits privileged EXEC mode.

Troubleshooting Tips

Use the **debug ppp negotiation** command to verify and troubleshoot MRRU negotiation on multilink groups. Use the **show interface** command to verify MRRU negotiation on the interfaces.

For more information about configuring MRRU and MTU values, see the Wide-Area Networking Configuration Guide: Multilink PPP, Cisco IOS XE Release 3S.

Troubleshooting PPP

You can troubleshoot PPP reliable link by using the **debug lapb** command and the **debug ppp negotiations**, **debug ppp errors**, and **debug ppp packets** commands. You can determine whether Link Access Procedure, Balanced (LAPB) has been established on a connection by using the **show interface** command.

Monitoring and Maintaining PPP and MLP Interfaces

You can use the **show ppp multilink** command to display MLP bundle information.

For more information about configuring MLPPP interfaces, see the Wide-Area Networking Configuration Guide: Multilink PPP, Cisco IOS XE Release 3S.

CHAPTER 6

Transparent SONET or SDH over Packet (TSoP) Protocol

Table 6: Feature History

Feature Name	Release Information	Description
SFP-TS-OC3STM1-I and SFP-TS-OC12STM4-I Optics Support	Cisco IOS XE 17.13.1	SFP-TS-OC3STM1-I and SFP-TS-OC12STM4-I Optics support is extended to the following chassis: • A900-RSP2A-128 These TSOP smart SFPs can operate in the low latency mode and is used for rail signaling. **Note** The support for the ONS-SC-155-TSOP and ONS-SC-622-TSOP are deprecated and are replaced with the new TSoPs SFP-TS-OC3STM1-I and SFP-TS-OC12STM4-I optics, respectively. The software functionalities of the new TSoPs remain the same as the earlier TSoPs.

Note Transparent SONET or SDH over Packet (TSoP) Protocol is *not* supported on the Cisco ASR 900 RSP3 module.

The Transparent SONET or SDH over Packet (TSoP) protocol converts SONET or SDH TDM traffic to a packet stream. Operators can now transport SONET or SDH traffic across a packet network by simply adding the TSoP Smart SFP to any router or packet switch. With TSoP the SONET or SDH signal is forwarded transparently, maintaining its embedded payload structure, protection protocols and synchronization. This simplifies the configuration and service turn-up of SONET or SDH connections across the packet network.

Starting with Cisco IOS XE Bengaluru 17.6.1, you can configure the jitter buffer to the lowest value on the Cisco RSP2 module.

Prerequisites for TSoP

- Single mode optical fiber must be used to connect TSoP Smart SFP with the OC-3 port.
- The TSoP smart SFP pseudowire endpoints must use the same configuration parameters.

Restrictions for TSoP

- The TSoP smart SFP payload size is *not* configurable. The byte size is fixed at 810 bytes.
- The router *cannot* be synced with the TSoP Smart SFP clock.
- Only untagged encapsulation is supported.
- CFM (connectivity fault management) is *not* supported.
- Only QoS Default Experimental marking is supported.
- TSoP can guarantee a sub 100-millisecond convergence time on SSO.
- SSO is not supported on TSoP for STM-4 or OC-12 SFP due to hardware restriction.
- TSoP is not supported on the 10G ports.
- After setting the low dejitter buffer value and when you change it to a higher value, ensure that you perform shut or no shut of the interface or IM OIR. Then only the new changes are applied on the interface takes place.

Information About TSoP Smart SFP

TSoP Smart SFP is a special type of optical transceiver which provides solution to transparently encapsulate SDH or SONET bit streams into packet format, suitable for pseudowire transport over an ethernet network The TSoP psuedowires is manually configured or setup using PWE3 control protocol [RFC4447].

TSoP provides packetization, de-packetization, and clock recovery that translates the TDM bit stream to fixed size data blocks (810 octets), and vice verse.

TSoP follows the SAToP method described in [RFC4553] for pseudowire transport of E1/DS1, over a packet switched network. With TSoP, the entire OC-3 or STM-1 is encapsulated in a single circuit emulating pseudowire traffic, and is transported it to a single destination across the ethernet network.

Note The TSoP smart SFP is used on any of the front panel ports of the 8-port Gigabit Ethernet SFP Interface Module (8X1GE).

- The Smart SFP transceivers is compatible with the Small Form Factor Pluggable 20-pin Multi-Source Agreement (MSA).
- TSoP Smart SFP (PN: ONS-SC-155-TSOP) transports upto 155 Mbps, on a L1.1 (40km) optical data link.

Guidelines for TSoP Smart SFP

TSoP is compatible with the below SFPs supported on the OC-3 interface module. We recommend you use the specified attenuator:

- ONS-SI-155-I1—For 15km cable length, use 2 dB attenuator; short distance use 8 dB attenuator to avoid receiver overload.
- ONS-SI-155-L1—For 40km cable length, no attenuator; short distance use 10 dB attenuator to avoid receiver overload.
- ONS-SI-155-L2—For 40km cable length, use 2 dB attenuator; short distance use 10 dB attenuator to avoid receiver overload.

Note Multimode SFP is not supported with TSoP.

STM-4 TSoP is compatible with the below SFPs supported on the OC-12 interface module:

- ONS-SI-622-L2—For 40km cable length, use 2 dB attenuator; short distance use 10 dB attenuator to avoid receiver overload.
- ONS-SI-622-L1—For 40km cable length, no attenuator; short distance use 10 dB attenuator to avoid receiver overload.
- ONS-SI-622-I1—For 15km cable length, use 2 dB attenuator; short distance use 8 dB attenuator to avoid receiver overload.

Note The OC-12 Smart SFP (PN: ONS-SC-622-TSOP) is *not* supported in Cisco IOS XE Release 3.14S.

> **Note** Effective Cisco IOS XE Release 3.18, STM-4 TSoP is supported on ASR 900 RSP2 Module .

Configuring the Reference Clock

The reference clock for the TSoP is extracted from the network. You can extract the clock reference from either of the following:

- Ethernet physical interface
- Incoming TDM physical interface

> **Note** If TDM reference clock is configured, and you want to return to the Ethernet reference clock (default), use the **ssfpd tsop clock-source ethernet** command. Additionally, you can also use the **no ssfpd tsop clock-source** command to return the Ethernet reference clock (default).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies the Gigabit Ethernet interface for configuration and enters interface configuration mode.
Step 4	**ssfpd tsop clock-source** {**ethernet** | **tdm**} **Example:** `Device(config)# ssfpd tsop clock-source ethernet`	Configures the reference clock on the interface. • **ethernet**—Specifies the ethernet interface as clock source. Default is ethernet. • **tdm**—Specifies the TDM interface as clock source.

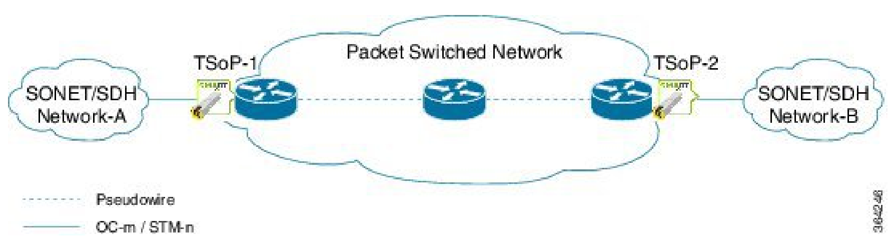
	Command or Action	Purpose
		Note If Ethernet interface is selected as clock source, the TSoP Smart SFP is synchronized with the Ethernet interface's clock (where smart SFP is installed), which in turn is synchronized with the network clock (that is already chosen through PTP or SYNC-E).
Step 5	**end** **Example:** `Device(config)# end`	Exists configuration and enters privileged EXEC mode

Configuration Examples for TSoP

Sample Configuration

For configuring SONET or SDH controller as in the figure (network A and B), see Configuring Optical Interface Modules.

Figure 10: TSoP in Packet Switched Network



TSoP Smart SFP inserted in the PE's, CE (SONET or SDH) can be configured as

- SDH or SONET framing for T1 and E1 mode.
- Serial interface in SDH or SONET mode. The scale for OC-3 IM is as supported—63 for E1 and 84 for T1 interfaces. The scale supported for OC-12 IM is 252 E1 and 336 T1 interfaces.
- Multilink interface with minimum of 1 member link and maximum of 16 member link.
- POS interface in SDH or SONET mode.
- ATM Layer3 interfaces in SDH or SONET mode.

Note ATM Layer 3 interface is not supported on CE for OC-12 IM.

- In OC-12 mode, if OC-12 IM is used on CE, only port 0 (ZERO) of the IM is used. Use the card-type command to operate the OC-12 IM.

For configuring the pseudowire using service instances, see Ethernet Virtual Connections Configuration on the Cisco ASR 903 Router.

Note Only untagged encapsulation is supported.

- The following example shows a sample configuration on the CE:

```
!
controller SONET 0/2/3
 framing sdh
 clock source line
 aug mapping au-3
 !
 !
 au-3 1
  overhead j1 length 64
  mode c-11
  tug-2 1 t1 1 channel-group 0 timeslots 1

!
```

- The following example shows a sample configuration of the Gigabit Ethernet interface with TSoP smart SFP installed:

```
!
interface GigabitEthernet0/0/0
 no ip address
 negotiation auto
 no keepalive
 service instance 1 ethernet
  encapsulation untagged
  xconnect 2.2.2.2 1 encapsulation mpls

!
```

Verification Examples

Verifying TSoP Smart SFP

- Use the **show inventory** command to display all TSoP Smart SFPs installed on the router.

```
Router# show inventory
NAME: "subslot 0/0 transceiver 7", DESCR: "TSoP OC-3/STM-1"
PID: ONS-SC-155-TSOP , VID: 01.0, SN: OES18100028
```

- Use the **show platform software ssfpd db** command to display all TSoP Smart SFPs recognized by the router.

```
Router# show  platform software ssfpd db
=== Smart SFP info ===
dpidx: 14
mac : 00:19:3a:00:2f:18
port: 7
bay: 0
```

```
ssfp upgrade data store id: -1
ssfp is device upgrade safe: -1
upgrade percentage complete: 0
ssfp upgrade in progress: 0
```

- Use the **show platform software ssfpd db** command with slot, bay and port to display specific TSoP Smart SFPs recognized by the router.

```
Router# show platform software ssfpd slot 0 bay 0 port 7 ssfp-d
port 7 ssfp-db
dpidx: 14
mac : 00:19:3a:00:2f:18
port: 7
bay: 0
ssfp upgrade data store id: -1
ssfp device upgrade safe: -1
Upgrade percentage_complete: 0
ssfp upgrade in progress: 0
```

- Use the **show hw-module subslot** command to view information about TSoP Smart SFP.

```
Router# show hw-module subslot 0/0 transceiver 7 idprom
IDPROM for transceiver GigabitEthernet0/0/7:
Description = SFP or SFP+ optics (type 3)
Transceiver Type: = TSoP OC-3/STM-1 (291)
Product Identifier (PID) = ONS-SC-155-TSOP
Vendor Revision = 01.0
Serial Number (SN) = OES18100028
Vendor Name = CISCO-OES
Vendor OUI (IEEE company ID) = 00.19.3A (6458)
CLEI code = WOTRDBZBAA
Cisco part number = 10-2949-01
Device State = Enabled.
Date code (yy/mm/dd) = 14/03/07
Connector type = LC.
Encoding = 8B10B
NRZ
Nominal bitrate = OC3/STM1 (200 Mbits/s)
```

The following example shows the configuration of STM-4 TSoP:

```
NAME: "subslot 0/5 transceiver 2", DESCR: "TSoP OC-12/STM-4"
PID: ONS-SC-622-TSOP , VID: 01.0, SN: OES17420029
```

The following example shows the configuration of ASR-920-12CZ-D:

```
router#show platform
Chassis type: ASR-920-12CZ-D

Slot      Type                State                 Insert time (ago)
--------- ------------------- --------------------- -----------------
 0/0      12xGE-2x10GE-FIXED  ok                    7w2d
R0        ASR-920-12CZ-D      ok, active            7w2d
F0                            ok, active            7w2d
P0        ASR920-PSU0         ps, fail              never
P1        ASR920-PSU1         ok                    never
P2        ASR920-FAN          ok                    never

Slot      CPLD Version        Firmware Version
--------- ------------------- ---------------------------------------
R0        20051844            15.6(53r)S
```

```
F0        20051844           15.6(53r)S
```

Verifying Clock Source

- Use the **show platform software ssfpd** command to display the configured clock source. In the following example, rtpClockSource value for Ethernet clock source is displayed as 0. For TDM clock source the rtpClockSource value is displayed as 1.

```
Router# show platform software ssfpd slot 0 bay 0 port 7 encap-params
sdId: 14
channel: 0
iwfEncapOutputEnable: 1
ecid: 0
gAisTriggerActive: 0
gAisIncludeLosTrigger: 1
gAisIncludeLofTrigger: 1
insertRtpHeader: 1
rtpClockSource: 0
rtpFrequency: 0
rtpPayloadType: 0
rtpSsrc: 0
```

Configuring the Low Dejitter Buffer

Table 7: Feature History

Feature Name	Release Information	Description
Support for TSoP SSFP Dejitter Buffer Tuning	Cisco IOS XE Bengaluru 17.6.1	TSoP SSFP Dejitter Buffer Tuning is applicable only for T1 smart SFP.

Dejitter buffer is a buffering mechanism to account for a delay variation in the CEM packet stream.

With the default value, the maximum size of the buffer is utilized. This means that the packets are delayed by default in the middle of the buffering mechanism. Depending on the quality of network, the dejitter buffer can be set to low or high.

If the network quality is high (low amount of packet-delay variation), then the size of the buffer can be set optionally to a low value using the **dejitter-buffer low** command.

Starting with Cisco IOS XE Bengaluru 17.6.1, you can configure jitter buffer to a lowest value.

To configure the low dejitter buffer, use the following commands:

```
configure terminal
interface <interface-type>  <slot/subslot/port>
ssfpd tsop dejitter-buffer low
end
```

```
Router#configure terminal
Router(config)#interface gigabitethernet 0/0/0
Router(config-if)#ssfpd tsop dejitter-buffer low
Router(config-if)#end
```

Use the following command to verify the configuration:

```
Router#show run | sec 0/3/5
interface GigabitEthernet0/3/5
no ip address
shutdown
negotiation auto
no keepalive
ssfpd tsop dejitter-buffer low
service instance 1 ethernet
encapsulation untagged
xconnect 10.10.10.10 111 encapsulation mpls
```