Configuring Pseudowire

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

Pseudowire Overview

The following sections provide an overview of pseudowire support on the router.

Effective Cisco IOS XE Release 3.18S:

- BGP PIC with TDM Pseudowire is supported on the ASR 900 router with RSP2 module.
- BGP PIC for Pseudowires, with MPLS Traffic Engineering is supported on the ASR 900 router with RSP1 and RSP2 modules.

Starting Cisco IOS XE Release 3.18.1SP, Pseudowire Uni-directional Active-Active is supported on the RSP1 and RSP3 modules.

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

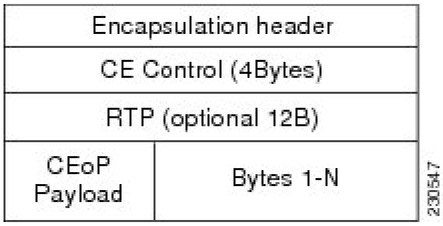
Structure-Agnostic TDM over Packet

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 3 shows the frame format in Unstructured SAToP mode.

Figure 1: Unstructured SAToP Mode Frame Format



#unique_5 unique_5_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_5 unique_5_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 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

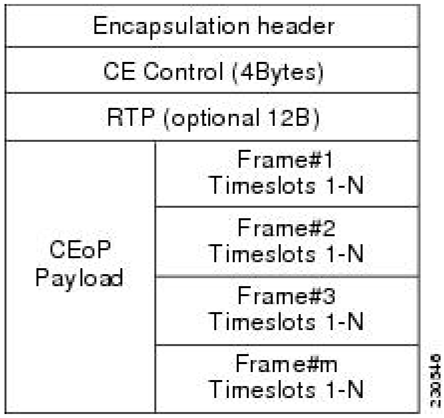


Table 3: CESoPSN DS0 Lines: Payload and Jitter Limits, on page 4 shows the payload and jitter for the DS0 lines in the CESoPSN mode.

Table 3: CESoPSN DS0 Lines: Payload and Jitter Limits

DS0	Maximum Payload	Maximum Jitter	Minimum Jitter	Minimum Payload	Maximum Jitter	Minimum Jitter
1	40	320	10	32	256	8
2	80	320	10	32	128	4
3	120	320	10	33	128	4
4	160	320	10	32	64	2
5	200	320	10	40	64	2
6	240	320	10	48	64	2
7	280	320	10	56	64	2
8	320	320	10	64	64	2
9	360	320	10	72	64	2
10	400	320	10	80	64	2
11	440	320	10	88	64	2
12	480	320	10	96	64	2
13	520	320	10	104	64	2
14	560	320	10	112	64	2
15	600	320	10	120	64	2
16	640	320	10	128	64	2

DS0	Maximum Payload	Maximum Jitter	Minimum Jitter	Minimum Payload	Maximum Jitter	Minimum Jitter
17	680	320	10	136	64	2
18	720	320	10	144	64	2
19	760	320	10	152	64	2
20	800	320	10	160	64	2
21	840	320	10	168	64	2
22	880	320	10	176	64	2
23	920	320	10	184	64	2
24	960	320	10	192	64	2
25	1000	320	10	200	64	2
26	1040	320	10	208	64	2
27	1080	320	10	216	64	2
28	1120	320	10	224	64	2
29	1160	320	10	232	64	2
30	1200	320	10	240	64	2
31	1240	320	10	248	64	2
32	1280	320	10	256	64	2

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 22.

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, the following limitation applies:

- BGP PIC with TDM Pseudowire is supported only on the ASR 900 router with RSP1 module.

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:** `Router(config-controller)# exit`	Exits the controller interface.

	Command or Action	Purpose
	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:** `Router(config)# exit`	Exits configuration mode.

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:
	`Router(config)# interface CEM 0/4`
	Example:
	`Router(config-if)# no ip address`
		Defines a CEM group.

	Command or Action	Purpose
	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.
Step 3	**controller [e1 | t1]** *slot/subslot* **Example:** `Router(config)# controller e1 0/0`	Enters configuration mode for the E1 or T1 controller.

	Command or Action	Purpose
	Example:	
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 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:**	Exits configuration mode.

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

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 1.1.1.1 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 1.1.1.1.
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.

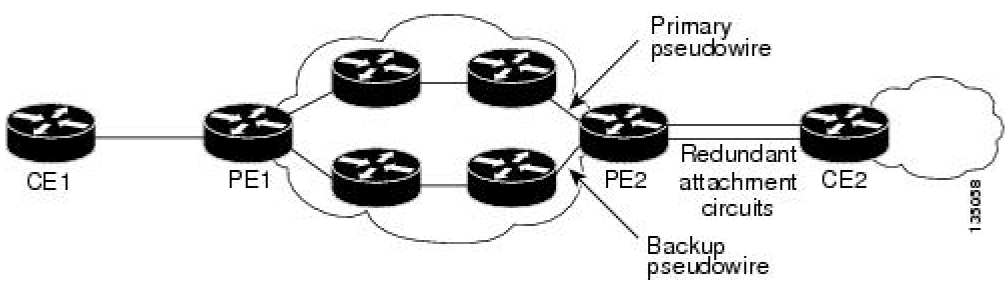
	Command or Action	Purpose
	`Router (config-if-srv)# xconnect 10.1.1.2 101 encapsulation mpls`	**Note** When creating IP routes for a pseudowire configuration, we recommend that you build a route from the xconnect address (LDP router-id or loopback address) to the next hop IP address, such as **ip route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

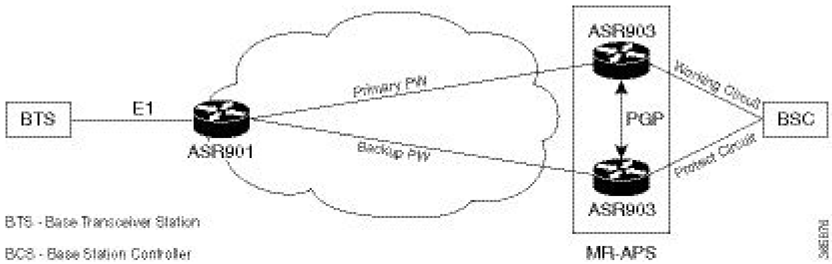
	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** [*pw-class-name*] **Example:** `Router(config)# pseudowire-class mpls`	Specify the name of a Layer 2 pseudowire class and enter pseudowire class 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 .

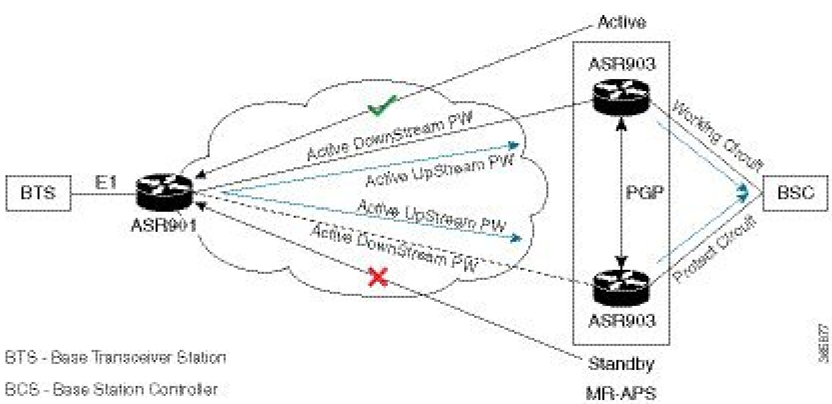
Figure 4: Pseudowire Redundancy with MR-APS



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 1.1.1.1 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

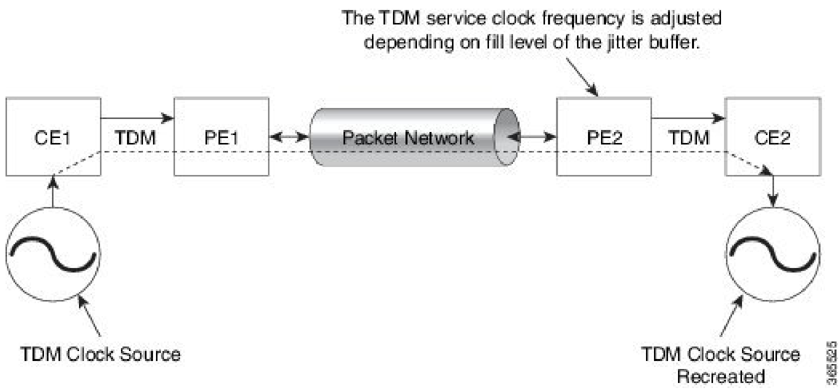
```
!
mpls label range 16 12000 static 12001 16000
mpls label protocol ldp
mpls ldp neighbor 10.5.5.5 targeted ldp
mpls ldp graceful-restart
multilink bundle-name authenticated
```

```
!
!
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