

Configuring Pseudowire

This chapter provides information about configuring pseudowire features on the Cisco ASR 920 Series Router.

- Pseudowire Overview, on page 1
- CEM Configuration, on page 2
- CEM Configuration Guidelines and Restrictions, on page 2
- Configuring a CEM Group, on page 3
- Using CEM Classes , on page 4
- Configuring CEM Parameters, on page 5
- Configuring Structure-Agnostic TDM over Packet (SAToP), on page 7
- Configuring Circuit Emulation Service over Packet-Switched Network (CESoPSN), on page 8
- Configuring an Ethernet over MPLS Pseudowire, on page 10
- Configuring Pseudowire Redundancy, on page 11
- Sample Configurations, on page 13

Pseudowire Overview

Effective Cisco IOS-XE Release 3.18S:

- BGP PIC with TDM Pseudowire is supported on the ASR 920 routers with RSP2 modules.
- BGP PIC for Pseudowires, with MPLS Traffic Engineering is supported on the ASR 920 router with RSP2 modules.

The following sections provide an overview of pseudowire support on the Cisco ASR 920 Series Router.

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.
- 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 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.
- If BGP PIC is enabled and the targeted LDP for VPWS Xconnect 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 xconnect 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.

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 920 Series Router implementation of EoMPLS PWs is compliant with the RFC 4447 and 4448 standards.

The Cisco ASR 920 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.

CEM Configuration

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.

Note Steps for configuring CEM features are also included in the Configuring Structure-Agnostic TDM over Packet (SAToP) and Configuring Circuit Emulation Service over Packet-Switched Network (CESoPSN), on page 8 sections.

CEM 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.

Configuring a CEM Group

The following section describes how to configure a CEM group on the Cisco ASR 920 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/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

	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	**class cem** *cem-class* **Example:** `Router(config)# class cem mycemclass`	Creates a new CEM class
Step 4	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	Router(config-cem-class)# exit	Returns to the config prompt.
Step 6	Router(config)# interface cem 0/0 **Example:**	Configure the CEM interface that you want to use for the new CEM class.

	Command or Action	Purpose

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

Step 7	Router(config-if-cem)# **exit** **Example:** `Router(config-if)#`	Exits the CEM interface.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits configuration mode.

Configuring CEM Parameters



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.

Custom Idle Pattern

Table 1: Feature History

Feature Name	Release Information	Description
Custom Idle Pattern	Cisco IOS XE Cupertino 17.9.1	You can configure idle pattern manually on CEM circuits and verify if it's stable and transmitted to the other end in alarm conditions. You can configure on all CEM PWs in a T1/E1 circuit. Supported on the following IMs on CESoPSN circuits with both partial and full time slots. • 48-port T1/E1 CEM Interface Module • 48-port T3/E3 CEM Interface Module These idle pattern numbers are used for tracking purposes.

To define the idle pattern that a circuit emulation (CEM) channel transmits when the channel experiences an underrun condition or to replace any missing packets, use the **idle-pattern** command in CEM configuration mode. Starting with Cisco IOS XE Cupertino 17.9.1 release, you can manually configure any 8-bit value from idle pattern. There are multiple CEMs in TDM circuits, these configurations are applicable only to CEM circuits.

For example, a controller T1 0/1/0, can have one CEM circuit. It’s only applicable for CESoP, the time slots can be 1–24, these are full time slots.

For example, under CEM0 you can manually configure any 8-bit value until 255 (0xFF). For partial time slot, consider CEM group 0 with time slot 0, and similarly CEM group 1 with time slot 1.

```
Router(config)# interface CEM0/10/10
Router(config-if)# cem 0
Router(config-if-cem)# idle-pattern 44
```

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
```

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 Structure-Agnostic TDM over Packet (SAToP)

Follow these steps to configure SAToP on the Cisco ASR 920 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.

	Command or Action	Purpose
Step 3	**controller [T1|E1] 0/4** **Example:** `Router(config-controller)# controller t1`	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	Router(config)# **interface CEM0/4**
	Example:
	`Router(config-if)# no ip address`
	Example:
	`Router(config-if)# cem 4`
		Defines a CEM group.

Step 6	Router(config-if)# **xconnect 30.30.30.2 304 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 30.30.2.304.
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 xconnect address (LDP router-id or loopback address) to the next hop IP address, such as **ip route 30.30.30.2 255.255.255.255 1.2.3.4.**

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

Follow these steps to configure CESoPSN on the Cisco ASR 920 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	Router(config)# **controller [e1|t1] 0/0** **Example:** `Router(config-controller)#`	Enters configuration mode for the E1 or T1 controller.
Step 4	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	Router(config-controller)# **exit** **Example:** `Router(config)#`	Exits controller configuration.
Step 6	Router(config)# **interface CEM0/5** **Example:** `Router(config-if-cem)# cem 5`	Defines a CEM channel.
Step 7	Router(config-if-cem)# **xconnect 30.30.30.2 305 encapsulation mpls**	Binds an attachment circuit to the CEM interface to create a pseudowire. This example creates a pseudowire by binding the CEM circuit 5 to the remote peer 30.30.30.2. **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 30.30.30.2 255.255.255.255 1.2.3.4.**
Step 8	Router(config-if-cem)# **exit** **Example:**	Exits the CEM interface.

	Command or Action	Purpose
	`Router(config)#`	
Step 9	**exit** **Example:** `Router(config)# exit`	Exits configuration mode.

Configuring an Ethernet over MPLS Pseudowire

Ethernet over MPLS PWs allow you to transport Ethernet traffic over an existing MPLS network. The Cisco ASR 920 Series 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 2. For more information about how to configure MPLS, see the Cisco IOS XE 3S Configuration Guides. For more information about configuring Ethernet Virtual Connections (EVCs), see Configuring Ethernet Virtual Connections on the Cisco ASR 920 Router.

Follow these steps to configure an Ethernet over MPLS Pseudowire on the Cisco ASR 920 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	**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

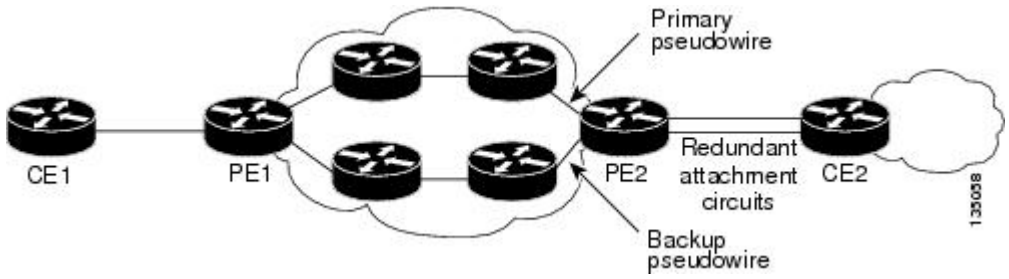
	Command or Action	Purpose
		need to use an EVC name in a service instance. **Note** You can use service instance settings such as encapsulation, dot1q, and rewrite to configure tagging properties for a specific traffic flow within a given pseudowire session. For more information, see Configuring Ethernet Virtual Connections on the Cisco ASR 920 Router.
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** {**mpls** [**manual**]} | **pw-class** *pw-class-name* }[**pw-class** *pw-class-name*] [**sequencing** {**transmit** | **receive** | **both**}] **Example:** `Router (config-if-srv)# xconnect 10.1.1.2 101 encapsulation mpls`	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. **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 10.30.30.2 255.255.255.255 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 920 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.

The figure below shows an example of pseudowire redundancy.

Figure 1: 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.
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	**gigabitethernet** *slot/port* **Example:** `Router(config)# gigabitethernet 0/0/1`	Enters configuration mode for the serial interface. **Note** The slot number is always 0.

	Command or Action	Purpose
Step 6	Router(config)# **backup delay** *enable-delay* {*disable-delay* | **never**}	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	Router(config-if)# **xconnect 1.1.1.2 101 encapsulation mpls**	Binds the Ethernet port interface to an attachment circuit to create a pseudowire.
Step 8	Router(config)# **backup peer** *peer-router-ip-address vcid* [**pw-class** *pw-class name*]	Defines the address and VC of the backup peer.
Step 9	**exit** **Example:** `Router(config)# exit`	Exits configuration mode.

Sample Configurations

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.

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: 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
!
```

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:

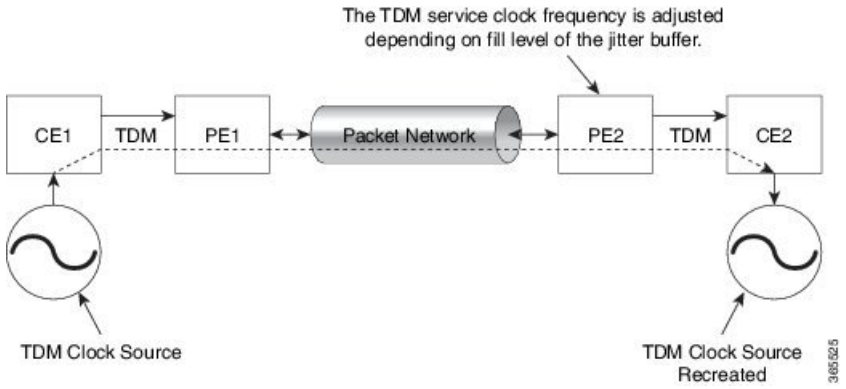
```
pseudowire-class pseudowire1
encapsulation mpls
control-word
no status control-plane route-watch
status peer topology dual-homed
!
Interface CEM0/0/0
cem 1
   xconnect 17.1.1.1 4101 encapsulation mpls pw-class pseudowire1
```

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.

Effective Cisco IOS XE Everest 16.5.1, ACR is supported on the 8-port T1/E1 interface module.



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

Commands	Links
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