Implementing IEEE 802.1ah Provider Backbone Bridge

This module provides conceptual and configuration information for IEEE 802.1ah Provider Backbone Bridge on Cisco ASR 9000 Series Routers. The IEEE 802.1ah standard (Ref [4]) provides a means for interconnecting multiple provider bridged networks to build a large scale end-to-end Layer 2 provider bridged network.

The Cisco ASR 9000 Series Aggregation Services Routers now supports a scenario when the provider backbone bridge is a VPLS network. You can now configure pseudowires in the PBB edge bridge domain and core bridge domain. In either type of bridge domain, the pseudowire functionality remains the same as in the native bridge domain.

Feature History for Implementing IEEE 802.1ah Provider Backbone Bridge

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.9.1</td>
<td>This feature was introduced on Cisco ASR 9000 Series Routers.</td>
</tr>
<tr>
<td>Release 4.3.0</td>
<td>Support was added for these features:</td>
</tr>
<tr>
<td></td>
<td>• Provider Backbone Bridge VPLS</td>
</tr>
<tr>
<td></td>
<td>• Multiple I-SID Registration Protocol Lite (MIRP Lite)</td>
</tr>
<tr>
<td>Release 4.3.2</td>
<td>Support was added for PBB-EVPN feature.</td>
</tr>
<tr>
<td>Release 5.1.2</td>
<td>Support was added for MMRP for PBB VPLS Flood Optimization feature.</td>
</tr>
</tbody>
</table>

Supported Hardware

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>ASR 9000 Ethernet Line Card</th>
<th>ASR 9000 Enhanced Ethernet Line Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic PBB</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Multiple I-SID Registration Protocol Lite</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>PBB VPLS</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>PBB EVPN</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MMRP for PBB VPLS Flood Optimization</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Prerequisites for Implementing 802.1ah Provider Backbone Bridge

This prerequisite applies to implementing 802.1ah Provider Backbone Bridge:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command.
  
  If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

- You must be familiar with the multipoint bridging concepts. Refer to the Implementing Multipoint Layer 2 Services module.

Information About Implementing 802.1ah Provider Backbone Bridge

To implement 802.1ah, you must understand these concepts:

- Benefits of IEEE 802.1ah standard, page 386
- IEEE 802.1ah Standard for Provider Backbone Bridging Overview, page 387
- Backbone Edge Bridges, page 388
- IB-BEB, page 389
- Multiple I-SID Registration Protocol Lite, page 390

Benefits of IEEE 802.1ah standard

The benefits of IEEE 802.1ah provider backbone bridges are as follows:

- Increased service instance scalability—Enables a service provider to scale the number of services (service VLANs or service instances) in a Provider Bridged Network (PBN).

- MAC address scalability—Encapsulates the customer packet, including MAC addresses, into a new ethernet frame with new MAC addresses (the backbone bridge MAC addresses). This eliminates the need for backbone core bridges to learn all MAC addresses of every customer and also eases the load on backbone edge bridges.
VPLS pseudowire reduction and mesh scalability—The number of pseudowires in an IP/MPLS core can be significantly reduced. This is because a single VPLS service can now transport several customer service instances thereby allowing a fewer number of pseudowires in the IP/MPLS core to transport a large number of customer services.

Layer 2 backbone traffic engineering—Enables explicit controls for Layer 2 traffic engineering by separating service discrimination function and moving it to the I-tags thereby leaving the backbone VLAN to be available for Layer 2 traffic engineering functions.

Point-to-point service scalability and optimization—Enables point-to-point service implementation that includes multiple options for service multiplexing as well as end point discovery.

Backbone flood traffic reduction—Since there are fewer MAC addresses in the core of the network, the amount of flood traffic in the core network is reduced as there are fewer MAC addresses to be relearnt when MAC tables get flushed due to topology changes.

IEEE 802.1ah Standard for Provider Backbone Bridging Overview

The IEEE 802.1ah Provider Backbone Bridge feature encapsulates or decapsulates end-user traffic on a Backbone Edge Bridge (BEB) at the edge of the Provider Backbone Bridged Network (PBBN). A Backbone Core Bridge (BCB) based network provides internal transport of the IEEE 802.1ah encapsulated frames within the PBBN. Figure 1 shows a typical 802.1ah PBB network.

![IEEE 802.1ah Provider Backbone Bridge](image)

**Figure 1  IEEE 802.1ah Provider Backbone Bridge**

PB - provider bridge
Figure 2 shows a typical provider backbone network topology.

**Figure 2 Provider Backbone Network Topology**

- Core can be Layer 2 or MPLS

**Backbone Edge Bridges**

Backbone edge bridges (BEBs) can contain either an I-Component or a B-Component. The I-Component maps service VLAN identifiers (S-VIDs) to service instance identifiers (I-SIDs) and adds a provider backbone bridge (PBB) header without a backbone VLAN tag (B-Tag). The B-Component maps I-SIDs to backbone VIDs (B-VIDs) and adds a PBB header with a B-Tag.

The IEEE 802.1ah standard specifies these three types of BEBs:

- The B-BEB contains the B-Component of the MAC-in-MAC bridge. It validates the I-SIDs and maps the frames onto the backbone VLAN (B-VLAN). It also switches traffic based on the B-VLANS within the core bridge.
The I-BEB contains the I-Component of the MAC-in-MAC bridge. It performs B-MAC encapsulation and inserts the I-SIDs based on the provider VLAN tags (S-tags), customer VLAN tags (C-tags), or S-tag/C-tag pairs.

The IB-BEB contains one or more I-Components and a single B-Component interconnected through a LAN segment.

**Note**

Only IB-BEBs are supported on Cisco ASR 9000 Series Routers. Cisco IOS XR supports IB-BEB bridge type at the Edge node.

**IB-BEB**

The IB-BEB contains both the I-Component and the B-Component. The bridge selects the B-MAC and inserts the I-SID based on the provider VLAN tag (S-tag), the customer VLAN tag (C-tag), or both the S-tag and the C-tag. It validates the I-SIDs and it transmits and receives frames on the B-VLAN.

The IEEE 802.1ah on Provider Backbone Bridges feature supports all services mandated by the IEEE 802.1ah standard and extends the services to provides these additional functionalities:

- **S-Tagged Service:**
  - In multiplexed environments each S-tag maps to an I-SID and may be retained or removed.
  - In bundled environments multiple S-tags map to the same I-SID and the S-tags must be retained.

- **C-Tagged Service:**
  - In multiplexed environments each C-tag maps to an I-SID and may be retained or removed.
  - In bundled environments multiple C-tags map to the same I-SID and the C-tags must be retained.

- **S/C-Tagged Service:**
  - In multiplexed environments each S-tag/C-tag pair maps to an I-SID. The S-tag or the S-tag/C-tag pair may be retained or removed.
  - In bundled environments multiple S-tag/C-tags pairs map to the same I-SID and the S-tag/C-tag pair must be retained.

- **Port-based Service**
  - A port-based service interface is delivered on a Customer Network Port (CNP). A port-based service interface may attach to a C-VLAN Bridge, 802.1d bridge, router or end-station. The service provided by this interface forwards all frames without an S-Tag over the backbone on a single backbone service instance. A port-based interface discards all frames with an S-Tag that have non-null VLAN IDs.

This example shows how to configure a port-based service:

```plaintext
interface GigabitEthernet0/0/0/10.100
  l2transport
  encapsulation untagged
--> Creates an EFP for untagged frames.

interface GigabitEthernet0/0/0/10.101
  l2transport
  encapsulation dot1ad priority-tagged
--> Creates an EFP for null S-tagged frames.

interface GigabitEthernet0/0/0/10.102
  l2transport
  encapsulation dot1q priority-tagged
--> Creates an EFP for null C-tagged frames:
```
Information About Implementing 802.1ah Provider Backbone Bridge

To configure a port-based service, all the above EFPs must be added to the same edge bridge domain.

Figure 3 shows the PBB bridge component topology on the Cisco ASR 9000 Series Routers.

Multiple I-SID Registration Protocol Lite

The 802.1Qbe—Multiple I-SID Registration Protocol (MIRP) standard provides the ability to flush learned MAC address registration entries held in the filtering database of an I-component on a per I-SID basis. The backbone service instance identifier (I-SID) is a field in the backbone service instance tag which identifies the backbone service instance of a frame. MIRP defines mechanisms for I-SID flushing, and has the required capabilities to handle topology changes that occur in networks attached to a provider backbone bridged network. A backbone edge bridge (BEB) signals to other potentially affected BEBs, the need to alter certain learned associations between customer MAC addresses and backbone MAC addresses. In the absence of MIRP, customer connections across a provider backbone network can take several minutes to restore connectivity after a topology change in an access network.
In prior releases, PBB traffic was dropped for a MAC aging cycle when bridge forwarding topology changes occurred (due to unavailable ports or spanning tree topology changes) in a PBB edge bridge domain. This resulted in severe limitations for the use of PBB bridges.

Cisco ASR 9000 Series Aggregation Services Routers now support a simplified implementation of the MIRP protocol known as the Multiple I-SID Registration Protocol Lite (MIRP-Lite). The MIRP-Lite feature enables detection of a topology change at a site. A specially defined packet is flooded to all remote edge sites of the PBB network when a site detects a topology change. At the sender site, I-SID of the I-component is placed in the I-TAG of the frame header to specify the I-SID that needs a MAC flush. At the receiver site, each PBB edge switch performs I-SID checking. If the I-SID matches one of the I-components, the MAC in the I-component is flushed.

The use of MIRP in 802.1ah networks is illustrated in Figure 4.
Device DHD1 is dual-homed to two 802.1ah backbone edge bridges (BEB1 and BEB2). Assume that initially the primary path is through BEB1. In this configuration BEB3 learns that the host behind DHD1 (with MAC address CM1) is reachable via the destination B-MAC M1. If the link between DHD1 and BEB1 fails and the host behind DHD1 remains inactive, the MAC cache tables on BEB3 still refer to the BEB1 MAC address even though the new path is now via BEB2 with B-MAC address M2. Any bridged traffic destined from the host behind DHD2 to the host behind DHD1 is wrongly encapsulated with B-MAC M1 and sent over the MAC tunnel to BEB1, where the traffic drops.

To circumvent the dropping of traffic when the link between DHD1 and BEB1 fails, BEB2 performs two tasks:

- Flushes its own MAC address table for the service or services.
- Requests the remote PE that receives the MIRP packet to clear its own MAC table. The MIRP message is transparent to the backbone core bridges (BCBs). The MIRP message is processed on a BEB because only BCBs learn and forward, based on B-MAC addresses and they are transparent to C-MAC addresses.

**Note**

MIRP triggers C-MAC address flushing for both native 802.1ah and PBB over VPLS.

*Figure 5* shows the operation of the MIRP.
Figure 5  MIRP Operation

Provider Backbone Bridging Ethernet VPN

The Provider Backbone Bridging Ethernet VPN (PBB-EVPN) is a next generation L2VPN solution that addresses resiliency and forwarding policy requirements. This feature also introduces advanced multihoming options, support for multipath and user-defined BGP policy capabilities to Ethernet L2VPNs. PBB-EVPN uses BGP for MAC address distribution and learning over the packet-switched network (PSN). PBB-EVPN is a combination of the capabilities of PBB and Ethernet VPN that addresses these Carrier Ethernet and data centre interconnect requirements:

- All-active Redundancy and Load Balancing
- Simplified Provisioning and Operation
- Optimal Forwarding
- Fast Convergence
- MAC Address Scalability
Ethernet VPN

Ethernet Virtual Private Network (EVPN) is a solution for secure and private connectivity of multiple sites within an organization. The EVPN service extends the benefits of Ethernet technology to the Wide Area Network (WAN). This service is delivered over MPLS networks.

EVPN allows you to manage routing over a virtual private network, providing complete control and security. EVPN introduces a solution for multipoint L2VPN services, with advanced multi-homing capabilities, using BGP for distributing customer or client MAC address reachability information over the MPLS/IP network. EVPN advertises each customer MAC address as BGP routes, therefore allowing BGP policy control over MAC addresses.

Figure 6 shows the MAC address distribution in BGP.

**Figure 6**  MAC Distribution in BGP (EVPN)

In Figure 6, the provider edge (PE) routers run multi-protocol BGP to advertise and learn MAC addresses over MPLS. The customer MAC addresses are learnt in the data plane over attachment circuits (links connecting customer devices to the PEs). Then, the MAC addresses are distributed over MPLS using BGP with an MPLS label identifying the EVPN instance.

**PBB-EVPN Overview**

The PBB-EVPN solution combines Ethernet Provider Backbone Bridging (PBB - IEEE 802.1ah) with Ethernet VPN where, PEs perform as PBB Backbone Edge Bridge (BEB). The PEs receive 802.1Q Ethernet frames from their attachment circuits. These frames are encapsulated in the PBB header and forwarded over the IP/MPLS core. On the egress side (EVPN PE), the PBB header is removed after MPLS disposition, and the original 802.1Q Ethernet frame is delivered to the customer equipment.
Figure 7 illustrates a PBB-EVPN network.

**Figure 7  PBB-EVPN Network**

The PE routers perform these functions:

- Learns customer or client MAC addresses (C-MACs) over the attachment circuits in the data-plane, per normal bridge operation.
- Learns remote C-MAC to backbone MAC (B-MAC) bindings in the data-plane from traffic ingress from the core.
- Advertises local B-MAC address reachability information in BGP to all other PE nodes in the same set of service instances. Note that every PE has a set of local B-MAC addresses that uniquely identify the device.
- Builds a forwarding table from the received remote BGP advertisements, associating remote B-MAC addresses with remote PE IP addresses.

PBB-EVPN scales well for large network with millions of customer MAC addresses by constraining customer MAC address in access. Only B-MAC addresses are advertised in core, making the number of BGP routes exchanged manageable.

**EVPN Instance**

E-VPN Instance (EVI) identifies a VPN in the MPLS/IP network. There can only be one EVI per core bridge.
**Ethernet Segment**

Ethernet Segment is a site connected to one or more PEs. The Ethernet Segment could be a single device (i.e., Customer Edge (CE)) or an entire network, such as:
- Single-Homed Device (SHD)
- Multi-Homed Device (MHD) using Ethernet Multi-chassis Link Aggregation Group
- Single-Homed Network (SHN)
- Multi-Homed Network (MHN)

The Ethernet segment is uniquely identified by a 10-byte global Ethernet Segment Identifier (ESI). Figure 8 illustrates an example of Ethernet segment and ESI.

**Figure 8 Ethernet Segment**

**PBB-EVPN BGP Routes**

PBB-EVPN defines a single new BGP network layer reachability information (NLRI) used to advertise different types of routes along with new attributes.

**Designated Forwarder Election**

The Designated Forwarder (DF) election mechanism is used to determine a designated forwarder in dual-homed or multi-homed devices or networks. The election is performed on a per service basis. The DF filtering function for MHN differs from that for MHD in:
- Directionality—DF filtering for MHN is applied for traffic both ingress and egress on the access-facing Ethernet interfaces; whereas, DF filtering for MHD is applied only to traffic that egress the access-facing interfaces.
- Traffic Type—DF filtering for MHN impacts both unicast as well as flooded multi-destination traffic; whereas, DF filtering for MHD only applies to flooded multi-destination traffic.

Figure 9 shows the various DF filtering rules for MHN and MHD.
Figure 9  DF Filtering Comparison for MHN/MHD

<table>
<thead>
<tr>
<th>Scenario</th>
<th>MHD w/Per Flow LB</th>
<th>MHN (always treated as SHN)</th>
<th>MHD w/Per service LB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filtering Direction</td>
<td>Egress</td>
<td>Egress</td>
<td>Egress</td>
</tr>
<tr>
<td>(on AC)</td>
<td></td>
<td>Ingress</td>
<td>Ingress</td>
</tr>
<tr>
<td>Filtered Traffic</td>
<td>Multicast</td>
<td>Multicast</td>
<td>Multicast</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unicast</td>
<td>Unicast</td>
</tr>
<tr>
<td>Granularity</td>
<td>EFP</td>
<td>EFP</td>
<td>EFP</td>
</tr>
</tbody>
</table>

Access Auto-Sensing

PEs connected to a multi-homed or dual-homed device may support active-active per flow also known as flow-based load balancing. PE services CEs via physical or bundle ports. An Ethernet segment identifier is assigned per port. This value is calculated from the connected CE using information such as, CE system priority, CE system ID and CE port key. The PE must auto-detect the access topology to determine the type of load balancing. The load balancing could be active-active per flow load-balancing, per service load-balancing or simply no load balancing.
## MMRP for PBB VPLS Flood Optimization

In a PBB network, traffic (unknown unicast, multicast, or broadcast) is flooded to all the PE devices in the network even if the devices do not host the service instance to which the traffic is destined.

The Multiple MAC Registration Protocol (MMRP) for PBB VPLS Flood Optimization feature optimizes the impact of the flooded traffic on PE devices by sending the traffic only to the PE devices interested in a particular service instance.

In a PBB over VPLS network, traffic between the PE devices flows over MPLS pseudo-wires that connect all the PE devices in a full mesh network topology.

**Figure 2** illustrates a typical 802.1ah PBB network.

For every I-SID (Service Instance VLAN ID) there is a corresponding multicast MAC address called the group B-MAC address, which is derived based on the I-SID. The group B-MAC address is used as the destination address in the outer MAC header when propagating flooded traffic across the provider backbone.

The MMRP is used by the PE devices to inform each other about the set of group B-MAC addresses corresponding to the I-SIDs of the service instances they host. This enables each device to determine which set of pseudo-wires flooded traffic should be forwarded on, that is, those pseudo-wires on which an MMRP registration has been received for the group B-MAC address corresponding to the I-SID.

**Note**
The PBB-VPLS flood optimization feature is enabled only on PBB-VPLS network and not on PBB over Ethernet network.

## Configuring PBB-VPLS Flood Optimization

To configure the PBB-VPLS flood optimization feature, do the following:

- Enabling PBB-VPLS Flood Optimization on PBB Core Bridge, page 398.
- (Optional) Configuring Generic MRP Protocol Parameters, page 400.

## Enabling PBB-VPLS Flood Optimization on PBB Core Bridge

Perform this task to enable PBB-VPLS flood optimization on PBB core bridge.

**SUMMARY STEPS**

1. `configure`
2. `l2vpn`
3. `bridge group group-name`
4. `bridge-domain domain-name`
5. `pbb core`
6. `mmrp-flood-optimization`
7. `end`
   or
    `commit`
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>l2vpn</code></td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# l2vpn</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><code>bridge group bridge-group-name</code></td>
<td>Enters configuration mode for the named bridge group. This command creates a new bridge group or modifies the existing bridge group if it already exists. A bridge group organizes bridge domains.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn)#bridge group pbb</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>bridge-domain domain-name</code></td>
<td>Enters configuration mode for the named bridge domain. This command creates a new bridge domain or modifies the existing bridge domain if it already exists.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg)#bridge-domain pbb-core</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><code>pbb core</code></td>
<td>Configures the bridge domain as PBB core and enters the PBB core configuration submode. This command also creates an internal port known as Customer bridge port (CBP). All the interfaces (bridge ports) under this bridge domain are treated as the provider network ports (PNP).</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# pbb core</td>
<td></td>
</tr>
</tbody>
</table>
Information About Implementing 802.1ah Provider Backbone Bridge

Configuring Generic MRP Protocol Parameters

Perform this task to configure the generic MRP protocol parameters for PBB-VPLS flood optimization.

SUMMARY STEPS

1. configure
2. mmrp-flood-optimization
3. (Optional) periodic transmit [interval seconds]
4. (Optional) join-time milliseconds
5. (Optional) leaveall-time seconds
6. (Optional) leave-time seconds
7. (Optional) flood-time seconds
8. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>mmrp-flood-optimization</td>
<td>Enables the flood optimization for PBB over VPLS feature on the core bridge.</td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-core)# mmrp-flood-optimization
```

Step 7

Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-core)# end
```

or

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-core)# commit
```

Saves configuration changes.

- When you issue the `end` command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)?

  [cancel]:

  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

  • Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.

When you issue the `commit` command, the system prompts you to commit changes:
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td><code>mmrp-flood-optimization</code></td>
<td>Enables the flood optimization for PBB over VPLS feature on the core bridge.</td>
</tr>
<tr>
<td>3</td>
<td><code>periodic transmit interval seconds</code></td>
<td>Enables periodic Multiple MAC Registration Protocol Data Units (MMRPDUs).</td>
</tr>
<tr>
<td>4</td>
<td><code>join-time milliseconds</code></td>
<td>Sets the join time for all active ports.</td>
</tr>
<tr>
<td>5</td>
<td><code>leaveall-time seconds</code></td>
<td>Sets the leave all time for all active ports.</td>
</tr>
<tr>
<td>6</td>
<td><code>leave-time seconds</code></td>
<td>Sets the leave time for all active ports.</td>
</tr>
</tbody>
</table>

**Example:**

- **Step 1:**
  ```
  RP/0/RSP0/CPU0:router# configure
  ```

- **Step 2:**
  ```
  RP/0/RSP0/CPU0:router(config)# mmrp-flood-optimization
  ```

- **Step 3:**
  ```
  RP/0/RSP0/CPU0:router(config-mmrp-flood-opt)# periodic transmit interval 3
  ```

- **Step 4:**
  ```
  RP/0/RSP0/CPU0:router(config-mmrp-flood-opt)# join time interval 300
  ```

- **Step 5:**
  ```
  RP/0/RSP0/CPU0:router(config-mmrp-flood-opt)# leaveall-time 10
  ```

- **Step 6:**
  ```
  RP/0/RSP0/CPU0:router(config-mmrp-flood-opt)# leave-time 40
  ```
### Information About Implementing 802.1ah Provider Backbone Bridge

#### Step 7

**flood-time** *seconds*

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Enables flooding of traffic to the entire core bridge when the PBB-VPLS Flood Optimization feature is enabled on the core bridge.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mmrp-flood-opt)# flood-time 1000</td>
<td></td>
</tr>
</tbody>
</table>

#### Step 8

**end**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-mmrp-flood-opt)# end</td>
<td>• When you issue the <strong>end</strong> command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td>or</td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)?</td>
</tr>
<tr>
<td>commit</td>
<td>[cancel]:</td>
</tr>
<tr>
<td></td>
<td>- Entering <strong>yes</strong> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>- Entering <strong>no</strong> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>- Entering <strong>cancel</strong> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>commit</strong> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>
How to Implement 802.1ah Provider Backbone Bridge

This section contains these procedures:

- Restrictions for Implementing 802.1ah Provider Backbone Bridge, page 403
- Configuring Ethernet Flow Points on CNP and PNP Ports, page 404
- Configuring PBB Edge Bridge Domain and Service Instance ID, page 406
- Configuring the PBB Core Bridge Domain, page 408
- Configuring Backbone VLAN Tag under the PBB Core Bridge Domain, page 409
- Configuring Backbone Source MAC Address, page 411 (optional)
- Configuring Unknown Unicast Backbone MAC under PBB Edge Bridge Domain, page 414 (optional)
- Configuring Static MAC addresses under PBB Edge Bridge Domain, page 416 (optional)
- Configuring PBB VPLS, page 417

Restrictions for Implementing 802.1ah Provider Backbone Bridge

These features are not supported:

- Cross-connect based point to point services over MAC-in-MAC
- One Edge bridge to multiple Core bridge mapping
- I type backbone edge bridge (I-BEB) and B type backbone edge bridge (B-BEB)
- IEEE 802.1ah over VPLS
- Multiple source B-MAC addresses per chassis
- Direct encapsulation of 802.1ah formatted packets natively over an MPLS LSP encapsulation

The following additional restriction applies when implementing Provider Backbone Bridge Ethernet VPN (PBB-EVPN):

- The Provider Edge and Route Reflector routers must run software supporting the same IETF draft version of L2VPN Ethernet VPN (EVPN). Due to the differences in BGP Network Layer Reachability Information (NLRI) encoding, later draft versions are not backward compatible with earlier ones. The following table shows the supported draft for various Cisco IOS XR software releases.

<table>
<thead>
<tr>
<th>Cisco IOS XR software release</th>
<th>Supported L2VPN EVPN draft version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>draft-ietf-l2vpn-evpn-04</td>
</tr>
<tr>
<td>5.1.1 and older releases</td>
<td>Yes</td>
</tr>
<tr>
<td>5.2.0</td>
<td>Yes</td>
</tr>
<tr>
<td>5.1.2 and later releases except 5.2.0</td>
<td>—</td>
</tr>
</tbody>
</table>
Configuring Ethernet Flow Points on CNP and PNP Ports

Perform this task to configure an Ethernet flow point (EFP) on the customer network port (CNP) or the provider network port (PNP).

SUMMARY STEPS

1. configure
2. interface type interface-path-id.subinterface l2transport
3. encapsulation dot1q vlan-id
   or
   encapsulation dot1ad vlan-id
   or
   encapsulation dot1ad vlan-id dot1q vlan-id
4. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**
```plaintext
RP/0/RSP0/CPU0:router# configure
```

<table>
<thead>
<tr>
<th><strong>Step 2</strong> interface type interface-path-id.subinterface l2transport</th>
<th>Configures an interface for L2 switching.</th>
</tr>
</thead>
</table>

**Example:**
```plaintext
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/0/0/10.100 l2transport
```
### Command or Action

**Step 3**

- `encapsulation dot1q vlan-id`
  or
- `encapsulation dot1ad vlan-id`
  or
- `encapsulation dot1ad vlan-id dot1q vlan-id`

**Example:**

```
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
or
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1ad 100
or
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1ad 100 dot1q 101
```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 4**
  
  - `end`
  - `commit`
  
  **Example:**
  
  ```
  RP/0/RSP0/CPU0:router(config-subif)# end
  or
  RP/0/RSP0/CPU0:router(config-subif)# commit
  ```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 4**
  
  - `end`
  - `commit`
  
  **Example:**
  
  ```
  RP/0/RSP0/CPU0:router(config-subif)# end
  or
  RP/0/RSP0/CPU0:router(config-subif)# commit
  ```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 4**
  
  - `end`
  - `commit`
  
  **Example:**
  
  ```
  RP/0/RSP0/CPU0:router(config-subif)# end
  or
  RP/0/RSP0/CPU0:router(config-subif)# commit
  ```

Saves configuration changes.

- When you issue the `end` command, the system prompts you to commit changes:  
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?  
  [cancel]:
  
  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring PBB Edge Bridge Domain and Service Instance ID

Perform this task to configure a PBB edge domain and the service ID.

*Note* To configure the PBB feature, login with admin user privileges and issue the `hw-module profile feature l2` command to select an ASR 9000 Ethernet line card ucode version that supports the PBB feature. The PBB feature will not be supported on the ASR 9000 Ethernet line card unless you make this configuration. For more information on configuring the feature profile, refer to the *Cisco ASR 9000 Series Aggregation Services Router System Management Configuration Guide*.

**SUMMARY STEPS**

1. `configure`
2. `l2vpn`
3. `bridge group group-name`
4. `bridge-domain domain-name`
5. `interface type interface-path-id.subinterface`
6. `pbb edge i-sid service-id core-bridge core-bridge-name`
7. `end`
   or
   `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
   `configure` | Enters global configuration mode. |
| **Example:**
   `RP/0/RSP0/CPU0:router# configure` | |
| **Step 2**
   `l2vpn` | Enters L2VPN configuration mode. |
| **Example:**
   `RP/0/RSP0/CPU0:router(config)# l2vpn` | |
| **Step 3**
   `bridge group bridge-group-name` | Enters configuration mode for the named bridge group. This command creates a new bridge group or modifies the existing bridge group if it already exists. A bridge group organizes bridge domains. |
| **Example:**
   `RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group pbb` | |
| **Step 4**
   `bridge-domain domain-name` | Enters configuration mode for the named bridge domain. This command creates a new bridge domain or modifies the existing bridge domain, if it already exists. |
| **Example:**
   `RP/0/RSP0/CPU0:router(config-l2vpn-bg)#bridge-domain pbb-edge` | |
## Command or Action

<table>
<thead>
<tr>
<th>Step 5</th>
<th>interface type interface-path-id.subinterface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#interface GigabitEthernet0/5/0/0.20</td>
</tr>
</tbody>
</table>

Assigns the matching VLAN ID and Ethertype to the interface. This EFP is considered as the CNP for the Edge bridge.

<table>
<thead>
<tr>
<th>Step 6</th>
<th>pbb edge i-sid service-id core-bridge core-bridge-name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#pbb edge i-sid 1000 core-bridge pbb-core</td>
</tr>
</tbody>
</table>

Configures the bridge domain as PBB edge with the service identifier and the assigned core bridge domain, and enters the PBB edge configuration submode.

This command also creates the Virtual instance port (VIP) that associates the PBB Edge bridge domain to the specified Core bridge domain.

All the interfaces (bridge ports) under this bridge domain are treated as the customer network ports (CNP).

<table>
<thead>
<tr>
<th>Step 7</th>
<th>end</th>
<th>or</th>
<th>commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-edge)# end</td>
<td>or</td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-edge)# commit</td>
</tr>
</tbody>
</table>

Saves configuration changes.

- When you issue the `end` command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
# Configuring the PBB Core Bridge Domain

Perform this task to configure the PBB core bridge domain.

## SUMMARY STEPS

1. configure
2. l2vpn
3. bridge group *group-name*
4. bridge-domain *domain-name*
5. interface type *interface-path-id.subinterface*
6. pbb core
7. end
   or
   commit

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> l2vpn</td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# l2vpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bridge group <em>bridge-group-name</em></td>
<td>Enters configuration mode for the named bridge group. This command creates a new bridge group or modifies the existing bridge group, if it already exists. A bridge group organizes bridge domains.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group pbb</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> bridge-domain <em>domain-name</em></td>
<td>Enters configuration mode for the named bridge domain. This command creates a new bridge domain or modifies the existing bridge domain if it already exists.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-l2vpn-bd)# bridge-domain pbb-core</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> interface type <em>interface-path-id.subinterface</em></td>
<td>Assigns the matching VLAN ID and Ethertype to the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-l2vpn-bd-bd)# interface GigabitEthernet0/5/0/0.20</td>
<td></td>
</tr>
</tbody>
</table>
How to Implement 802.1ah Provider Backbone Bridge

Chapter Implementing IEEE 802.1ah Provider Backbone Bridge

How to Implement 802.1ah Provider Backbone Bridge

Configuring Backbone VLAN Tag under the PBB Core Bridge Domain

Perform this task to configure the backbone VLAN tag under the PBB core bridge domain.

SUMMARY STEPS

1. configure
2. l2vpn
3. bridge group group-name
4. bridge-domain domain-name
5. interface type interface-path-id.subinterface
6. interface type interface-path-id.subinterface
7. pbb core
8. rewrite ingress tag push dot1ad vlan-id symmetric
9. end
   or
   commit

Step 6

Configure the bridge domain as PBB core and enters the PBB core configuration submode.

This command also creates an internal port known as Customer bridge port (CBP).

All the interfaces (bridge ports) under this bridge domain are treated as the provider network ports (PNP).

Example:

RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# pbb core

Step 7

Saves configuration changes.

- When you issue the end command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Example:

RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-core)# end
or
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-core)# commit
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> l2vpn</td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# l2vpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bridge-group bridge-group-name</td>
<td>Enters configuration mode for the named bridge group. This command creates a new bridge group or modifies the existing bridge group if it already exists. A bridge group organizes bridge domains.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group pbb</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> bridge-domain domain-name</td>
<td>Enters configuration mode for the named bridge domain. This command creates a new bridge domain or modifies the existing bridge domain if it already exists.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain pbb-core</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> interface type interface-path-id.subinterface</td>
<td>Assigns the matching VLAN ID and Ethertype to the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface GigabitEthernet0/5/0/0.20</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> interface type interface-path-id.subinterface</td>
<td>Adds an interface to a bridge domain that allows packets to be forwarded and received from other interfaces that are part of the same bridge domain. The interface now becomes an attachment circuit on this bridge domain.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# interface GigabitEthernet0/5/0/1.15</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> pbb core</td>
<td>Configures the bridge domain as PBB core and enters the PBB core configuration submode. This command also creates an internal port known as Customer bridge port (CBP). All the interfaces (bridge ports) under this bridge domain are treated as the provider network ports (PNP).</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# pbb core</td>
<td></td>
</tr>
</tbody>
</table>
### Step 8

**Command or Action**

```plaintext
rewrite ingress tag push dot1ad vlan-id symmetric
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-core)# end
```

**Purpose**

Configures the backbone VLAN tag in the Mac-in-MAC frame and also, sets the tag rewriting policy.

**Note**

All PNs in a Core bridge domain use the same backbone VLAN.

### Step 9

**Command or Action**

```plaintext
end
```

**Purpose**

Saves configuration changes.

- When you issue the `end` command, the system prompts you to commit changes:
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?
  [cancel]:
  
  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.

---

### Configuring Backbone Source MAC Address

The backbone source MAC address (B-SA) is a unique address for a backbone network. Each Cisco ASR 9000 Series Router has one backbone source MAC address. If B-SA is not configured, then the largest MAC in the EEPROM is used as the PBB B-SA.

**Note**

The backbone source MAC address configuration is optional. If you do not configure the backbone source MAC address, the Cisco ASR 9000 Series Routers allocate a default backbone source MAC address from the chassis backplane MAC pool.
Perform this task to configure the backbone source MAC address.

**SUMMARY STEPS**

1. configure
2. l2vpn
3. pbb
4. backbone-source-mac *mac-address*
5. end
   or
5. commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> l2vpn</td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# l2vpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> pbb</td>
<td>Enters PBB configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-l2vpn)# pbb</td>
<td></td>
</tr>
</tbody>
</table>
## Implementing IEEE 802.1ah Provider Backbone Bridge

### How to Implement 802.1ah Provider Backbone Bridge

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td><code>backbone-source-address mac-address</code></td>
<td>Configures the backbone source MAC address.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-pbb)# backbone-source-address 0045.1200.04</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td><code>end</code> or <code>commit</code></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-pbb)# end or RP/0/RSP0/CPU0:router(config-l2vpn-pbb)# commit</td>
<td></td>
</tr>
</tbody>
</table>

- When you issue the `end` command, the system prompts you to commit changes:
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
  
  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring Unknown Unicast Backbone MAC under PBB Edge Bridge Domain

Perform this task to configure the unknown unicast backbone MAC under the PBB edge bridge domain.

**SUMMARY STEPS**

1. configure
2. l2vpn
3. bridge group group-name
4. bridge-domain domain-name
5. interface type interface-path-id.subinterface
6. pbb edge i-sid service-id core-bridge core-bridge-name
7. unknown-unicast-bmac mac-address
8. end
   or
   commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>l2vpn</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config)# l2vpn</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>bridge group bridge-group-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters configuration mode for the named bridge group. This command creates a new bridge group or modifies the existing bridge group if it already exists. A bridge group organizes bridge domains.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn)#bridge group pbb</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>bridge-domain domain-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Enters configuration mode for the named bridge domain. This command creates a new bridge domain or modifies the existing bridge domain if it already exists.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg)#bridge-domain pbb-edge</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>interface type interface-path-id.subinterface</td>
</tr>
<tr>
<td>Example:</td>
<td>Assigns the matching VLAN ID and Ethertype to the interface.</td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#interface GigabitEthernet0/5/0/0.20</td>
</tr>
</tbody>
</table>
### How to Implement 802.1ah Provider Backbone Bridge

#### Step 6

**Command or Action:**

```
pbb edge i-sid service-id core-bridge core-bridge-name
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# pbb edge i-sid 1000 core-bridge pbb-core
```

**Purpose:**

Configures the bridge domain as PBB edge with the service identifier and the assigned core bridge domain and enters the PBB edge configuration submode.

This command also creates the Virtual instance port (VIP) that associates the PBB Edge bridge domain to the specified Core bridge domain.

All the interfaces (bridge ports) under this bridge domain are treated as the customer network ports (CNP).

#### Step 7

**Command or Action:**

```
unknown-unicast-bmac mac-address
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-edge)# unknown-unicast-bmac 1.1.1
```

**Purpose:**

Configures unknown unicast backbone MAC address.

**Note**

On Trident line cards, once you configure the unknown unicast BMAC, the BMAC is used to forward customer traffic with multicast, broadcast and unknown unicast destination MAC address.

#### Step 8

**Command or Action:**

```
end
```

or

```
commit
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-edge)# end
```

or

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-edge)# commit
```

**Purpose:**

Saves configuration changes.

- When you issue the `end` command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring Static MAC addresses under PBB Edge Bridge Domain

Perform this task to configure the static MAC addresses under the PBB edge bridge domain.

SUMMARY STEPS

1. configure
2. l2vpn
3. bridge group group-name
4. bridge-domain domain-name
5. interface type interface-path-id.subinterface
6. interface type interface-path-id.subinterface
7. pbb edge i-sid service-id core-bridge core-bridge-name
8. static-mac-address cda-mac-address bmac bda-mac-address
9. end
or
commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td>l2vpn</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# l2vpn</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters configuration mode for the named bridge group. This command creates a new bridge group or modifies the existing bridge group if it already exists. A bridge group organizes bridge domains.</td>
</tr>
<tr>
<td>bridge group bridge-group-name</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group pbb</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Enters configuration mode for the named bridge domain. This command creates a new bridge domain or modifies the existing bridge domain if it already exists.</td>
</tr>
<tr>
<td>bridge-domain domain-name</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain pbb-edge</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Assigns the matching VLAN ID and Ethertype to the interface.</td>
</tr>
<tr>
<td>interface type interface-path-id.subinterface</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface GigabitEthernet0/5/0/0.20</td>
</tr>
</tbody>
</table>
How to Implement 802.1ah Provider Backbone Bridge

Configuring PBB VPLS

Perform these tasks to configure PBB VPLS:

- Configuring Access Pseudowire in I-Component, page 418
- Configuring Core Pseudowire in B-Component, page 420

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 6**
**interface** type interface-path-id.subinterface | Adds an interface to a bridge domain that allows packets to be forwarded and received from other interfaces that are part of the same bridge domain. The interface now becomes an attachment circuit on this bridge domain. |
| Example: RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)#interface GigabitEthernet0/5/0/1.15 | |
| **Step 7**
**pbb edge i-sid** service-id core-bridge core-bridge-name | Configures the bridge domain as PBB edge with the service identifier and the assigned core bridge domain and enters the PBB edge configuration submode. |
| Example: RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#pbb edge i-sid 1000 core-bridge pbb-core | This command also creates the Virtual instance port (VIP) that associates the PBB Edge bridge domain to the specified Core bridge domain. All the interfaces (bridge ports) under this bridge domain are treated as the customer network ports (CNP). |
| **Step 8**
**static-mac-address** cda-mac-address bmac bda-mac-address | Configures the static CMAC to BMAC mapping under the PBB Edge submode. |
| Example: RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-edge)#static-mac-address 0033.3333.3333 bmac 0044.4444.4444 | |
| **Step 9**
**end** or **commit** | Saves configuration changes. |
| Example: RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-edge)# end or RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-edge)# commit | When you issue the **end** command, the system prompts you to commit changes: |
| | Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: |
| | – Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. |
| | – Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. |
| | – Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. |
| | • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |
Chapter
Implementing IEEE 802.1ah Provider Backbone Bridge

Configuring Access Pseudowire in I-Component

Perform this task to configure the static MAC addresses under the PBB edge bridge domain.

SUMMARY STEPS

1. configure
2. l2vpn
3. bridge group group-name
4. bridge-domain domain-name
5. mac withdraw state-down
6. exit
7. interface type interface-path-id.subinterface
8. interface type interface-path-id.subinterface
9. neighbor {A.B.C.D} pw-id value
10. exit
11. pbb edge i-sid service-id core-bridge core-bridge-name
12. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 l2vpn</td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# l2vpn</td>
<td></td>
</tr>
<tr>
<td>Step 3 bridge group bridge-group-name</td>
<td>Enters bridge group configuration mode. This command creates a new bridge group or modifies the existing bridge group if it already exists. A bridge group organizes bridge domains.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group pbb</td>
<td></td>
</tr>
<tr>
<td>Step 4 bridge-domain domain-name</td>
<td>Enters bridge domain configuration mode. This command creates a new bridge domain or modifies the existing bridge domain if it already exists.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain pbb-edge</td>
<td></td>
</tr>
</tbody>
</table>
### How to Implement 802.1ah Provider Backbone Bridge

**Step 5**

**Command or Action**

`mac withdraw state-down`

**Purpose**

(Optional) Enables MAC withdrawal.

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn/bg-bd)# mac withdraw state-down
```

**Step 6**

**Command or Action**

`exit`

**Purpose**

Exits the current configuration mode.

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn/bg-bd-mac)# exit
```

**Step 7**

**Command or Action**

`interface type interface-path-id.subinterface`

**Purpose**

Assigns the matching VLAN ID and Ethertype to the interface.

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn/bg-bd)# interface GigabitEthernet0/5/0/0.20
```

**Step 8**

**Command or Action**

`interface type interface-path-id.subinterface`

**Purpose**

Adds an interface to a bridge domain that allows packets to be forwarded and received from other interfaces that are part of the same bridge domain. The interface now becomes an attachment circuit on this bridge domain.

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn/bg-bd-ac)# interface GigabitEthernet0/5/0/1.15
```

**Step 9**

**Command or Action**

`neighbor {A.B.C.D} pw-id value`

**Purpose**

Adds an access pseudowire port to a bridge domain.

- Use the `A.B.C.D` argument to specify the IP address of the cross-connect peer.

**Note**

- `A.B.C.D` can be a recursive or non-recursive prefix.
- Use the `pw-id` keyword to configure the pseudowire ID and ID value. The range is 1 to 4294967295.

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn/bg-bd)# neighbor 10.1.1.2 pw-id 1000
RP/0/RSP0/CPU0:router(config-l2vpn/bg-bd-pw)#
```

**Step 10**

**Command or Action**

`exit`

**Purpose**

Exits the current configuration mode.

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn/bg-bd-pw)# exit
```
How to Implement 802.1ah Provider Backbone Bridge

Perform this task to configure the static MAC addresses under the PBB edge bridge domain.

**SUMMARY STEPS**

1. configure
2. l2vpn
3. bridge group group-name
4. bridge-domain domain-name
5. vfi {vfi-name}
6. neighbor {A.B.C.D} {pw-id value}
7. end
   or
   commit

**DETAILED STEPS**

**Command or Action**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 11 pbb edge i-sid service-id core-bridge core-bridge-name</td>
<td>Configures the bridge domain as PBB edge with the service identifier and the assigned core bridge domain and enters the PBB edge configuration submode. All the interfaces (bridge ports) under this bridge domain are treated as the customer network ports (CNP).</td>
</tr>
</tbody>
</table>

**Example:**

RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# pbb edge i-sid 1000 core-bridge pbb-core

**Step 12 end**

or

commit

**Example:**

RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-edge)# end

or

RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-edge)# commit

Saves configuration changes.

- When you issue the end command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

**Configuring Core Pseudowire in B-Component**

Perform this task to configure the static MAC addresses under the PBB edge bridge domain.
## How to Implement IEEE 802.1ah Provider Backbone Bridge

### Purpose

- Enters global configuration mode.
- Enters L2VPN configuration mode.
- Enters configuration mode for the named bridge group. This command creates a new bridge group or modifies the existing bridge group if it already exists. A bridge group organizes bridge domains.
- Enters configuration mode for the named bridge domain. This command creates a new bridge domain or modifies the existing bridge domain if it already exists.
- Configures the virtual forwarding interface (VFI) parameters and enters L2VPN bridge group bridge domain VFI configuration mode.
  - Use the `vfi-name` argument to configure the name of the specified virtual forwarding interface.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>l2vpn</code></td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RSP0/CPU0:router(config)# l2vpn</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>bridge group bridge-group-name</code></td>
<td>Enters configuration mode for the named bridge group. This command creates a new bridge group or modifies the existing bridge group if it already exists. A bridge group organizes bridge domains.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group pbb</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>bridge-domain domain-name</code></td>
<td>Enters configuration mode for the named bridge domain. This command creates a new bridge domain or modifies the existing bridge domain if it already exists.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain pbb-core</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>vfi (vfi-name)</code></td>
<td>Configures the virtual forwarding interface (VFI) parameters and enters L2VPN bridge group bridge domain VFI configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# vfi PBB-core-vfi</code></td>
<td></td>
</tr>
</tbody>
</table>
How to Implement 802.1ah Provider Backbone Bridge

Step 6

**neighbor (A.B.C.D) {pw-id value}**

**Example:**
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#
neighbor 10.1.1.2 pw-id 1000
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pw)#

- Adds an access pseudowire port to a bridge domain or a pseudowire to a bridge virtual forwarding interface (VFI).
- Use the `A.B.C.D` argument to specify the IP address of the cross-connect peer.

**Note**  `A.B.C.D` can be a recursive or non-recursive prefix.
- Use the `pw-id` keyword to configure the pseudowire ID and ID value. The range is 1 to 4294967295.

Step 7

**end**
or

**commit**

**Example:**
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pw)# end
or
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pw)# commit

- Saves configuration changes.
- When you issue the `end` command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring PBB-EVPN

This section provides information on:
- Configuring PBB Core Bridge Domains
- Configuring PBB Edge Bridge Domains
- Configuring EVPN Ethernet Segment
- Configuring BGP Route Target
- Configuring Global EVPN Timers
- Configuring EVPN Timers Per Ethernet Segment and CE flushing mechanism
- Configuring Multichassis Link Aggregation
- Configuring BGP Routing Process

Configuring PBB Core Bridge Domains

Perform this task to create the PBB Core bridge domain and assign its corresponding EVPN EVI ID.

SUMMARY STEPS

1. configure
2. l2vpn
3. bridge group group_name
4. bridge-domain bridge_domain_name
5. pbb core
6. evpn evi evi_id
7. end
   or
   commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td>l2vpn</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# l2vpn</td>
<td></td>
</tr>
</tbody>
</table>
## How to Implement 802.1ah Provider Backbone Bridge

### Step 3
**bridge group group_name**

**Example:**
```
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group group1
```

Enters bridge group configuration mode. This command creates a new bridge group. A bridge group organizes bridge domains.

### Step 4
**bridge-domain bridge_domain_name**

**Example:**
```
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain sample-pbb-core
```

Enters bridge group domain configuration mode. This command creates a new bridge domain.

### Step 5
**pbb core**

**Example:**
```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# pbb core
```

Configures the bridge domain as PBB core and enters the PBB core configuration submode. This command also creates an internal port known as Customer bridge port (CBP). All the interfaces (bridge ports) under this bridge domain are treated as the provider network ports (PNP).

### Step 6
**evpn evi evi_id**

**Example:**
```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-core)# evpn evi 100
```

Enters EVPN configuration mode and configures the Ethernet VPN ID. The EVI ID range is from 1 to 65534.

### Step 7
**end** or **commit**

**Example:**
```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-core)# end
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
```

  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring PBB Edge Bridge Domains

As a pre-requisite, a PBB-EVPN provider edge (PE) must be configured with PBB Edge Bridge Domains which in one side are associated to ethernet flow points matching traffic from access interfaces and on the other side are linked to PBB Core Bridge Domains for traffic forwarding through the core.

For more information on configuring Edge Bridge Domains, see Configuring PBB Edge Bridge Domain and Service Instance ID.

Configuring EVPN Ethernet Segment

Explicit configuration of Ethernet Segment parameters such as ESI and service carving behaviors (manual or dynamic) is required only for Dual Homed scenarios with Active/Active per Service load-balancing.

**Note**

By default, Dual Homed scenarios with Active/Active per Flow load-balancing auto-sense ESI values from CE’s LACP information.

**Note**

PBB-EVPN configuration allows to create only 24 ICCP-groups.

Perform this task to configure the EVPN Ethernet segment.

**SUMMARY STEPS**

1. configure
2. evpn
3. interface type interface-path-id
4. ethernet-segment
5. backbone-source-mac H.H.H
6. force single-homed
7. identifier system-priority priority-value system-id system-id port-key port_key
8. load-balancing-mode per-service
9. service-carving manual primary {isid} secondary {isid}
10. end
   or
   commit
## How to Implement 802.1ah Provider Backbone Bridge

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>evpn</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config)# evpn</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>interface type interface-path-id</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-evpn)# interface gigabitEthernet 0/1/0/4</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>ethernet-segment</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-evpn-ac)# ethernet-segment</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>backbone-source-mac mac_address</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-evpn-ac)# backbone-source-mac 0045.1200.04</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>force single-homed</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-evpn-ac-es)# force single-homed</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>identifier system-priority priority-value system-id system-id port-key port_key</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-evpn-ac-es)# identifier system-priority 2 system-id 0301.1.1 port-key ffff</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>load-balancing-mode per-service</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-evpn-ac-es)# load-balancing-mode per-service</td>
</tr>
</tbody>
</table>
**Configuring BGP Route Target**

By default, these parameters are auto-derived from the PE’s configuration:

- **Route Distinguisher (RD)** for global Ethernet Segment table
  
  Default: Auto-generated RD based on loopback IP address

- **EVI’s BGP Route Distinguisher (RD)**
  
  Default: Auto-generated RD based on loopback IP address

- **EVI’s BGP Route Target.** Default: Auto-generated RT based on EVI ID

Perform this task to overwrite the auto-generated BGP RD/RT values.

**SUMMARY STEPS**

1. configure
2. evpn
3. bgp
4. rd \{2-byte as\_number \| 4-byte as\_number \| IP\_address \| none\} : \{nn\}
5. exit
6. evpn

---

### Configuring BGP Route Target

**Step 9**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>service-carving manual primary (isid) secondary (isid)</td>
<td>Specifies a list of service identifiers (isid) as active and standby services. The <strong>isid</strong> range is from 256 to 16777216.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config-evpn-ac-es)# service-carving manual primary 100 secondary 200
```

**Step 10**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>end or commit</td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config-evpn-ac-es)# end or commit
```

- When you issue the **end** command, the system prompts you to commit changes:
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
How to Implement 802.1ah Provider Backbone Bridge

- **evi evi_id**
- **bgp**
- **route-target [import | export] {2-byte as_number | 4-byte as_number | IP_address | none} : {nn}**
- **rd {2-byte as_number | 4-byte as_number | IP_address | none} : {nn}**
- **end**
  - or
  - **commit**

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> evpn</td>
<td>Enters EVPN configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# evpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bgp</td>
<td>Enters EVPN BGP configuration mode and configures static BGP settings for the Ethernet Segment ES:GLOBAL EVI, which is used for handling ES routes.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-evpn)# bgp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> rd {2-byte as_number</td>
<td>4-byte as_number</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-evpn-bgp)# rd 200:50</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Returns to the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-evpn-bgp)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> evpn</td>
<td>Enters EVPN configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# evpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> evi evi_id</td>
<td>Configures Ethernet VPN ID.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-evpn)# evi 100</td>
<td>The EVI ID range is from 1 to 65534.</td>
</tr>
<tr>
<td><strong>Step 8</strong> bgp</td>
<td>Enters the BGP configuration mode for the specific EVI.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-evpn-evi)# bgp</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Global EVPN Timers

Perform this task to configure global EVPN timers.

**SUMMARY STEPS**

1. configure
2. evpn
3. timers [flushagain | peering | programming | recovery]
4. end
   or
   commit

**Command or Action**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>route-target {import</td>
<td>export} (2-byte as_number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The import keyword imports routing information from the target VPN extended community.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The export keyword exports routing information to the target VPN extended community.</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-evpn-evi-bgp)# route-target 10:20</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>rd (2-byte as_number</td>
<td>4-byte as_number</td>
</tr>
<tr>
<td></td>
<td>Example: RP/0/RSP0/CPU0:router(config-evpn-evi-bgp)# rd 25:30</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>end or commit</td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>
|      | Example: RP/0/RSP0/CPU0:router(config-evpn-evi-bgp)# end or RP/0/RSP0/CPU0:router(config-evpn-evi-bgp)# commit | • When you issue the end command, the system prompts you to commit changes:
|      |             | Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: |
|      |             | • Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. |
|      |             | • Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. |
|      |             | • Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. |
|      |             | • Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session. |
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**
```
RP/0/RSP0/CPU0:router# configure
```

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>evpn</td>
<td>Enters the EVPN configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**
```
RP/0/RSP0/CPU0:router(config)# evpn
```
### How to Implement 802.1ah Provider Backbone Bridge

#### Step 3

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`timers [flushagain</td>
<td>peering</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config-evpn)# timers flushagain 40
```

- **Flush-again timer (for AAPS only):** When a MAC flush is sent, usually at the end of the programming timer expiration, a flush-again timer is started for the flush-again timer value. When it expires, another MAC flush message (MVRP or STP-TCN) is sent to the CE. This timer can be configured per segment-interface.
  - Range: 0 to 120 seconds, 0 means disabled
  - Default: 60 seconds

- **Peering timer:** Once all conditions are met to advertise to BGP, the PE waits for the peering timer value before advertising its RT, ESI and, Local MAC if it is Single-Home.
  - Range: 0 to 300 seconds, 0 means disabled
  - Default: 45 seconds

- **Programming timer:** Indicated time required by the HW to apply the carving results. At the end of the programming timer expiration, the next Ethernet Segment route object will be processed.
  - Range: 0 to 100000 microseconds
  - Default: 1500 microseconds

- **Recovery timer (for AAPS only):** Once the interface is up, the PE waits for the recovery timer value in order to allow the CE running STP protocol to converge. This timer can be configured per segment-interface.
  - Range: 20 to 3600 seconds
  - Default: 20 seconds

**Note:** Changing timers is only useful for scale configurations.
How to Implement 802.1ah Provider Backbone Bridge

Configuring EVPN Timers Per Ethernet Segment and CE flushing mechanism

Perform this task to configure per Ethernet segment timers.

SUMMARY STEPS

1. configure
2. evpn
3. interface type interface-path-id
4. ethernet-segment
5. mac-flush mvrp
6. timers [flushagain | recovery]
7. end
   or
   commit

Command or Action | Purpose
--- | ---
end  
or
commit | Saves configuration changes.

* When you issue the end command, the system prompts you to commit changes:

Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
- Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
- Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
- Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.
* Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Example:
RP/0/RSP0/CPU0:router(config-evpn-timers)# end
RP/0/RSP0/CPU0:router(config-evpn-timers)# commit
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> evpn</td>
<td>Enters the EVPN configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# evpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> interface type interface-path-id</td>
<td>Enters the physical port interface or the bundle interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-evpn)# interface gigabitEthernet 0/1/0/4</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ethernet-segment</td>
<td>Enters the EVPN ethernet-segment configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-evpn-ac)# ethernet-segment</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> mac-flush mvrp</td>
<td>Specifies MAC flush mode for this Ethernet Segment.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-evpn-ac)#</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Multichassis Link Aggregation

Multichassis Link Aggregation (MCLAG) is used in scenarios involving Multi Homed Devices. You must create an ICCP redundancy group in order to specify relevant MLACP parameters, such as, `mlacp system mac`, `mlacp system priority`, `mlacp node id` and backbone interfaces.

### Command or Action

**Step 6**

```bash
 timers [flushagain | recovery]
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-evpn-ac)# timers flushagain 40
```

**Purpose**

Configures per Ethernet segment timers.

- **Flush-again timer (for AApS only):** When a MAC flush is sent, usually at the end of the programming timer expiration, a flush-again timer is started for the flush-again timer value. When it expires, another MAC flush message (MVRP or STP-TCN) is sent to the CE. This timer can be configured per segment-interface.
  
  Range: 0 to 120 seconds, 0 means disabled
  
  Default: 60 seconds

- **Recovery timer (for AApS only):** Once the interface is up, the PE waits for the recovery timer value in order to allow the CE running STP protocol to converge. This timer can be configured per segment-interface.
  
  Range: 20 to 3600 seconds
  
  Default: 20 seconds

**Note** Changing timers is only useful for scale configurations.

**Step 7**

```bash
 end
```

**or**

```bash
 commit
```

**Example:**

```
RP/0/RSP0/CPU0:router(config-evpn-ac-timers)# end
```

**Purpose**

Saves configuration changes.

- **When you issue the `end` command, the system prompts you to commit changes:**
  
  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
  
  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
**Note**

Even though the redundancy group is created under the `redundancy-iccp-group` sub-mode, the solution does not rely on an actual ICCP session between PEs connected to the same site. The mode singleton command has been introduced to alert ICCP module.

For more information on configuring MCLAG, refer to the *Configuring Link Bundling on the Cisco ASR 9000 Series Router* module in the *Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide*.

### Configuring BGP Routing Process

A prerequisite of PBB-EVPN involves enabling the new EVPN address family under the BGP routing process and under BGP neighbor submode. For more information on BGP, refer to the *Implementing BGP* module in the *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

Perform this task to enable EVPN address family under BGP routing process and BGP neighbor submode.

#### SUMMARY STEPS

1. `configure`
2. `router bgp asn_id`
3. `address-family l2vpn evpn`
4. `exit`
5. `neighbor peer_ip_add`
6. `address-family l2vpn evpn`
7. `end` or `commit`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>router bgp asn_id</code></td>
<td>Specifies the BGP AS number and enters the BGP configuration mode, allowing you to configure the BGP routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RSP0/CPU0:router(config)# router bgp 100</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>address-family l2vpn evpn</code></td>
<td>Enables EVPN address family under BGP routing process and enters EVPN address family configuration submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RSP0/CPU0:router(config-bgp)# address-family l2vpn evpn</code></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 4**

`exit`

*Example:*

```
RP/0/RSP0/CPU0:router(config-bgp)# exit
```

Exits the current configuration mode.

**Step 5**

`neighbor peer_ip_add`

*Example:*

```
RP/0/RSP0/CPU0:router(config-bgp)# neighbor 10.1.1.1
```

Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.

**Step 6**

`address-family l2vpn evpn`

*Example:*

```
RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family l2vpn evpn
```

Enables EVPN address family under BGP routing process and enters EVPN address family configuration submode.

**Step 7**

`end`

or

`commit`

*Example:*

```
RP/0/RSP0/CPU0:router(config-bgp-nbr-af)# end
```

or

```
RP/0/RSP0/CPU0:router(config-bgp-nbr-af)# commit
```

Saves configuration changes.

- When you issue the `end` command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)?

  [cancel]:

  - Entering `yes` saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.
Chapter: Implementing IEEE 802.1ah Provider Backbone Bridge

Configuration Examples for Implementing 802.1ah Provider Backbone Bridge

This section provides these configuration examples:

- Configuring Ethernet Flow Points: Example, page 437
- Configuring PBB Edge Bridge Domain and Service Instance ID: Example, page 437
- Configuring PBB Core Bridge Domain: Example, page 438
- Configuring Backbone VLAN Tag: Example, page 438
- Configuring Backbone Source MAC Address: Example, page 438
- Configuring Static Mapping and Unknown Unicast MAC Address under the PBB Edge Bridge Domain, page 439
- Configuring PBB-VPLS: Example, page 439
- Configuring MIRP Lite: Example, page 440

Configuring Ethernet Flow Points: Example

This example shows how to configure Ethernet flow points:

```
config
interface GigabitEthernet0/0/0/10.100 l2transport
en encapsulation dot1q 100
or
  encapsulation dot1ad 100
or
  encapsulation dot1ad 100 dot1q 101
```

Configuring PBB Edge Bridge Domain and Service Instance ID: Example

This example shows how to configure the PBB edge bridge domain:

```
config
l2vpn
  bridge group PBB
  bridge-domain PBB-EDGE
    interface GigabitEthernet0/0/0/38.100
    interface GigabitEthernet0/2/0/30.150
    pbb edge i-sid 1000 core-bridge PBB-CORE
    !
    !
```
Configuring PBB Core Bridge Domain: Example

This example shows how to configure the PBB core bridge domain:

```plaintext
config
l2vpn
  bridge group PBB
  bridge-domain PBB-CORE
  interface G0/5/0/10.100
! interface G0/2/0/20.200
! pbb core
!
```

Configuring Backbone VLAN Tag: Example

This example shows how to configure the backbone VLAN tag:

```plaintext
config
l2vpn
  bridge group PBB
  bridge-domain PBB-CORE
  interface G0/5/0/10.100
! interface G0/2/0/20.200
! pbb core
    rewrite ingress tag push dot1ad 100 symmetric
!
```

Configuring Backbone Source MAC Address: Example

This example shows how to configure the backbone source MAC address:

```plaintext
config
l2vpn
  pbb
    backbone-source-mac 0045.1200.04
!
```
Configuring Static Mapping and Unknown Unicast MAC Address under the PBB Edge Bridge Domain

This example shows how to configure static mapping and unknown unicast MAC address under the PBB edge bridge domain:

```
config
l2vpn
bridge group PBB
  bridge-domain PBB-EDGE
    interface GigabitEthernet0/0/0/38.100
    !
    interface GigabitEthernet0/2/0/30.150
    !
    pbb edge i-sid 1000 core-bridge PBB-CORE
    static-mac-address 0033.3333.3333 bmac 0044.4444.4444
    unknown-unicast-bmac 0123.8888.8888
    !
```

Configuring PBB-VPLS: Example

This example shows you how to configure PBB VPLS.

**Configuring Access Pseudowire in I-component**

```
l2vpn
  bridge group PBB
    bridge-domain PBB-EDGE
      mac withdraw state-down ------ can be used with MIRP, optional
      interface GigabitEthernet0/0/0/38.100
      interface GigabitEthernet0/2/0/30.150
      neighbor 10.10.10.1 pw-id 1010 ------- configures access PW
      !
      pbb edge i-sid 1200 core-bridge PBB-CORE
      !
```

**Configuring Core Pseudowire in B-component**

```
l2vpn
  bridge group PBB
    bridge-domain PBB-CORE
      interface G0/5/0/10.100
      !
      vfi PBB-CORE-vfi
      neighbor 1.1.1.1 pw-id 1004 ------- configures core PW
      !
```


Configuring MIRP Lite: Example

The MIRP feature is enabled by default. However, MIRP packets are sent when the attachment circuit is not functional and you have configured `mac withdraw state-down` as shown:

```
12vpn
    bridge group PBB
    bridge-domain PBB-EDGE
    mac withdraw state-down
```

However, if you have not configured `mac withdraw state-down`, then MIRP packets are sent when the attachment circuit is functional.
## Additional References

These sections provide references related to implementing 802.1ah on Cisco ASR 9000 Series Routers.

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.1ah commands: complete command syntax, command modes, command history,</td>
<td>Provider Backbone Bridge Commands module in Cisco ASR 9000 Series Aggregation</td>
</tr>
<tr>
<td>defaults, usage guidelines, and examples</td>
<td>Services Router VPN and Ethernet Services command reference</td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support</td>
<td>—</td>
</tr>
<tr>
<td>for existing standards has not been modified by this feature.</td>
<td></td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco</td>
</tr>
<tr>
<td></td>
<td>MIB Locator found at this URL and choose a platform under the Cisco Access</td>
</tr>
<tr>
<td></td>
<td>Products menu:</td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support</td>
<td>—</td>
</tr>
<tr>
<td>for existing RFCs has not been modified by this feature.</td>
<td></td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>