

## In This Chapter

This section provides information about Provider Backbone Bridging (PBB) — Ethernet Virtual Leased Line in an MPLS-based network which is applicable to all of the 7750 SR, 7450 ESS and 7710 SR routers.

Topics in this section include:

- [Applicability on page 490](#)
- [Overview on page 491](#)
- [Configuration on page 493](#)
- [Conclusion on page 513](#)

## **Applicability**

This section is applicable to all 7750 SR, 7450 ESS and 7710 SR series and was tested on release 12.0.R5. There are no specific prerequisites required.

## Overview

The draft-ietf-l2vpn-pbb-vpls-pe-model-00, *Extensions to VPLS PE model for Provider Backbone Bridging*, describes the PBB-VPLS model supported by SR OS. This model expands the VPLS PE model to support PBB as defined by the IEEE 802.1ah.

The PBB model is organized around a B-component (backbone instance) and an I-component (customer instance). In Alcatel-Lucent's implementation of the PBB model, the use of an Epipe as I-component is allowed for point-to-point services. Multiple I-VPLS and Epipe services can be all mapped to the same B-VPLS (backbone VPLS instance).

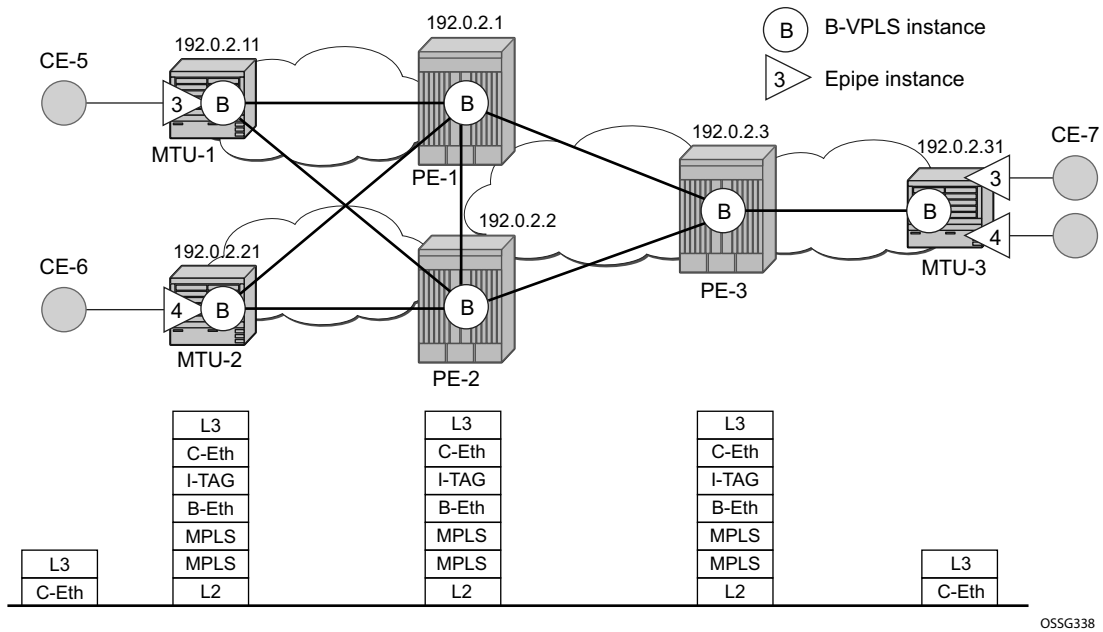
The use of Epipe scales the E-Line services as no MAC switching, learning or replication is required in order to deliver the point-to-point service. All packets ingressing the customer SAP are PBB-encapsulated and unicasted through the B-VPLS tunnel using the backbone destination MAC of the remote PBB PE. All the packets ingressing the B-VPLS destined for the Epipe are PBB de-encapsulated and forwarded to the customer SAP.

Some use cases for PBB-Epipe are:

- Get a more efficient and scalable solution for point-to-point services:
  - Up to 8K VPLS services per box are supported (including I-VPLS or B-VPLS) and using I-VPLS for point-to-point services takes VPLS resources as well as unnecessary customer MAC learning. A better solution is to connect a PBB-Epipe to a B-VPLS instance, where there is no customer MAC switching/learning.
- Take advantage of the pseudowire aggregation in the M:1 model:
  - Many Epipe services may use only a single service and set of pseudowires over the backbone.
- Have a uniform provisioning model for both point-to-point (Epipe) and multipoint (VPLS) services.
  - Using the PBB-Epipe, the core MPLS/pseudowire infrastructure does not need to be modified: the new Epipe inherits the existing pseudowire and MPLS structure already configured on the B-VPLS and there is no need for configuring new tunnels or pseudowire switching instances at the core.

Knowledge of the PBB-VPLS architecture and functionality on the service router family is assumed throughout this section. For additional information, refer to the relevant Alcatel-Lucent user documentation.

The following network setup will be used throughout the rest of the chapter.



**Figure 74: Network Topology**

The setup consists of a three 7x50 SR/ESS (PE-1, PE-2 and PE-3) core and three Multi-Tenant Unit (MTU) nodes connected to the core. A backbone VPLS instance (B-VPLS 101) will be defined in all the six nodes, whereas two Epipe services will be defined as illustrated in Figure 74 (Epipe 3 in nodes MTU-1 and MTU-3, Epipe 4 in nodes MTU-2 and MTU-3). Those Epipe services will be multiplexed into the common B-VPLS 101, using the I-Service ID (ISID) field within the I-TAG as the demultiplexer field required at the egress MTU to differentiate each specific customer. Note that I-VPLS and Epipe services can be mapped to the same B-VPLS.

The B-VPLS domain constitutes a H-VPLS network itself, with spoke SDPs from the MTUs to the core PE layer. Active/standby (A/S) spoke SDPs can be used from the MTUs to the PEs (like in the MTU-1 and MTU-2 cases) or single non-redundant spoke SDPs (like MTU-3).

The protocol stack being used along the path between the CEs is represented in Figure 74.

## Configuration

This section describes all the relevant PBB-Epipe configuration tasks for the setup shown in [Figure 74](#). Note that the appropriate B-VPLS and associated IP/MPLS configuration is out of the scope of this document. In this particular example the following protocols will be configured beforehand in the core:

- ISIS-TE as IGP with all the interfaces being level-2. Alternatively OSPF could have been used.
- RSVP-TE as the MPLS protocol to signal the transport tunnels.
- LSPs between core PEs will be fast re-route protected (facility bypass tunnels) whereas LSP tunnels between MTUs and PEs will not be protected.
- The protection between MTU-1, MTU-2 and PE-1, PE-2 will be based on the A/S pseudowire protection configured in the B-VPLS.
- BGP is configured for auto-discovery, BGP-AD (Layer 2 VPN family), since FEC 129 will be used to establish the pseudowires between PEs in the core (FEC 128 between MTU and PE nodes).

Once the IP/MPLS infrastructure is up and running, the service configuration tasks described in the following sections can be implemented.

## PBB Epipe Service Configuration

In this particular example, the Epipes 3 and 4 are using the B-VPLS 101 in the core. The same B-VPLS which is multiplexing the Epipe services into a common service provider infrastructure can also be used to connect the I-VPLS instances existing in the network for multipoint services.

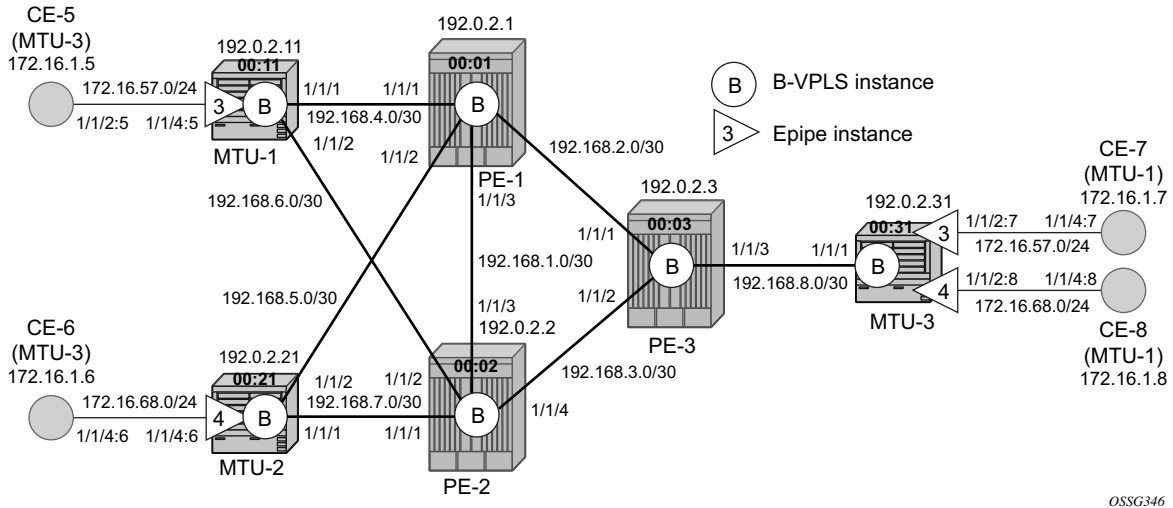


Figure 75: Setup Detailed View

OSSG346

## B-VPLS and PBB Configuration

First, configure the B-VPLS instance that will carry the PBB traffic. There is no specific requirement on the B-VPLS to support Epipes. The following shows the B-VPLS configuration on MTU-1 and PE-1.

```
# for MTU-1
configure
  service
    vpls 101 customer 1 b-vpls create
      service-mtu 2000
      pbb
        source-bmac 00:11:11:11:11:11
      exit
      stp
        shutdown
      exit
      endpoint "core" create
        no suppress-standby-signaling
      exit
      spoke-sdp 111:101 endpoint "core" create
        stp
          shutdown
        exit
        precedence primary
        no shutdown
      exit
      spoke-sdp 112:101 endpoint "core" create
        stp
          shutdown
        exit
        no shutdown
      exit
      no shutdown
    exit

# for PE-1
configure
  service
    pw-template 1 use-provisioned-sdp create
      split-horizon-group "CORE"
    exit
  exit
  vpls 101 customer 1 b-vpls create
    service-mtu 2000
    pbb
      source-bmac 00:01:01:01:01:01
    exit
    bgp
      route-target export target:65000:101 import target:65000:101
      pw-template-binding 1
    exit
  exit
  bgp-ad
    vpls-id 65000:101
```

## PBB Epipe Service Configuration

```
        no shutdown
    exit
    stp
        shutdown
    exit
    spoke-sdp 111:101 create
        no shutdown
    exit
    spoke-sdp 121:101 create
        no shutdown
    exit
    no shutdown
exit
```

The relevant B-VPLS commands are in **bold**.

Note that the keyword **b-vpls** is given at creation time and therefore it cannot be added to an existing regular VPLS instance. Besides the **b-vpls** keyword, the B-VPLS is a regular VPLS instance in terms of configuration, with the following exceptions:

- The B-VPLS service MTU must be at least 18 bytes greater than the Epipe MTU of the multiplexed instances. In this example, the I-VPLS instances will have the default service MTU (1514 bytes) hence any MTU equal or greater than 1532 bytes must be configured. In this particular example, a MTU of 2000 bytes is configured in the B-VPLS instance throughout the network.
- The source B-MAC is the MAC that will be used as a source when the PBB traffic is originated from that node. Note that you can configure a source B-MAC per B-VPLS instance (if there are more than one B-VPLS) or a common source B-MAC that will be shared by all the B-VPLS instances in the node. The way to configure a common B-MAC is shown below:

```
configure
  service
    pbb
      source-bmac 00:11:11:11:11:11
```

The following considerations will be taken into account when configuring the B-VPLS:

- B-VPLS SAPs:
  - Ethernet DOT1Q and NULL encapsulations are supported.
  - Default SAP types are blocked in the CLI for the B-VPLS SAP.



- B-VPLS SDPs:
  - For MPLS, both mesh and spoke SDPs with split horizon groups are supported.
  - Similar to regular pseudowire, the outgoing PBB frame on an SDP (for example, Bpseudowire) contains a BVID Qtag only if the pseudowire type is Ethernet VLAN (vc-type=vlan). If the pseudowire type is Ethernet (vc-type=ether), the BVID qtag is stripped before the frame goes out.
  - BGP-AD is supported in the B-VPLS, therefore, spoke SDPs in the B-VPLS can be signalled using FEC 128 or FEC 129. In this example, BGP-AD and FEC 129 are used. A split-horizon group has been configured to emulate the behavior of mesh SDPs in the core.
- While Multiple MAC Registration Protocol (MMRP) is useful to optimize the flooding in the B-VPLS domain and build a flooding tree on a per I-VPLS basis, it does not have any effect for Epipes since the destination B-MAC used for Epipes is always the destination B-MAC configured in the Epipe and never the group B-MAC corresponding to the ISID.
- If a local Epipe instance is associated with the B-VPLS, local frames originated or terminated on local Epipe(s) are PBB encapsulated or de-encapsulated using the PBB Etype provisioned under the related port or SDP component.

By default, the PBB Etype is 0x88e7 (which is the standard one defined in the 802.1ah, indicating that there is an I-TAG in the payload) but this PBB Etype can be changed if required due to interoperability reasons. This is the way to change it at port and/or SDP level:

```
A:MTU-1# configure port 1/1/1 ethernet pbb-etype
- pbb-etype <0x0600..0xffff>
- no pbb-etype

<0x0600..0xffff>      : [1536..65535] - accepts in decimal or hex

A:MTU-1# configure service sdp 111 pbb-etype
- no pbb-etype [<0x0600..0xffff>]
- pbb-etype <0x0600..0xffff>

<0x0600..0xffff>      : [1536..65535] - accepts in decimal or hex
```

The following commands are useful to check the actual PBB etype.

```
A:MTU-1# show service sdp 111 detail | match PBB
Bw BookingFactor      : 100                PBB Etype                : 0x88e7
A:MTU-1#

A:MTU-1# show port 1/1/1 | match PBB
PBB Ethertype       : 0x88e7
A:MTU-1#
```

## PBB Epipe Service Configuration

Before the next step, the Epipe configuration, the operator can optionally configure MAC names under the PBB context. MAC names will simplify the Epipe provisioning later on and in case of any change on the remote node MAC address, only one configuration modification is required as opposed as one change per affected Epipe (potentially thousands of Epipes which are terminated onto the same remote node). The MAC names are configured under the service PBB CLI context:

```
*A:MTU-1# configure service pbb mac-name
- mac-name <name> <ieee-address>
- no mac-name <name>

<name>                : 32 char max
<ieee-address>       : xx:xx:xx:xx:xx:xx or xx-xx-xx-xx-xx-xx

*A:MTU-1>config>service# info
-----
pbb
  source-bmac 00:11:11:11:11:11
  mac-name "MTU-1" 00:11:11:11:11:11
  mac-name "MTU-2" 00:21:21:21:21:21
  mac-name "MTU-3" 00:31:31:31:31:31
exit
```

## Epipe Configuration

Once the common B-VPLS is configured, the next step is the provisioning of the customer Epipe instances. For PBB-Epipes, the I-component or Epipe is composed of an I-SAP and a PBB tunnel endpoint which points to the backbone destination MAC address (B-DA).

The following outputs show the relevant CLI configuration for the two Epipe instances represented in [Figure 75 on page 494](#). The Epipe instances are configured on the MTU devices, whereas the core PEs are kept as customer-unaware nodes.

The following shows the relevant Epipe commands on MTU-3.

```
configure
  service
    pbb
      source-bmac 00:31:31:31:31:31
      mac-name "MTU-1" 00:11:11:11:11:11
      mac-name "MTU-2" 00:21:21:21:21:21
      mac-name "MTU-3" 00:31:31:31:31:31
    exit
  epipe 3 customer 1 create
    description "pbb epipe number 3"
    pbb
      tunnel 101 backbone-dest-mac "MTU-1" isid 3
    exit
    sap 1/1/2:7 create
    exit
    no shutdown
  exit
  epipe 4 customer 1 create
    description "pbb epipe number 4"
    pbb
      tunnel 101 backbone-dest-mac "MTU-2" isid 4
    exit
    sap 1/1/2:8 create
    exit
    no shutdown
  exit
```

It is not required to configure a node with its own MAC address, so the line defining the mac-name MTU-3 can be omitted.

## PBB Epipe Service Configuration

The following shows the relevant configuration on MTU-1 and MTU-2.

```
# for MTU-1
configure
  service
    epipe 3 customer 1 create
      description "pbb epipe number 3"
      pbb
        tunnel 101 backbone-dest-mac "MTU-3" isid 3
      exit
    sap 1/1/4:5 create
    exit
  no shutdown
exit

# for MTU-2
configure
  service
    epipe 4 customer 1 create
      description "pbb epipe number 4"
      pbb
        tunnel 101 backbone-dest-mac "MTU-3" isid 4
      exit
    sap 1/1/4:6 create
    exit
  no shutdown
exit
```

All Ethernet SAPs supported by a regular Epipe are also supported in the PBB Epipe. Note that spoke SDPs are not supported in PBB-Epipes, for example, no spoke SDP is allowed when PBB tunnels are configured on the Epipe.

The PBB tunnel links the SAP configured to the B-VPLS 101 existing in the core. The following parameters are accepted in the PBB tunnel configuration:

```
A:MTU-2>config>service>epipe>pbb# tunnel
- no tunnel
- tunnel <service-id> backbone-dest-mac <mac-name> isid <ISID>
- tunnel <service-id> backbone-dest-mac <ieee-address> isid <ISID>

<service-id>      : [1..2147483690] | <svc-name:64 char max>
<mac-name>       : 32 char max
<ieee-address>   : xx:xx:xx:xx:xx:xx or xx-xx-xx-xx-xx-xx
<ISID>          : [0..16777215]
```

Where:

- The service-id matches the B-VPLS ID.
- The **backbone-dest-mac** can be given by a MAC name (as in this configuration example) or the MAC itself. It is recommended to use MAC names, as explained in the previous section.
- The ISID must be specified.

## Flood Avoidance in PBB-Epipes

As already discussed in the previous section, when provisioning a PBB Epipe, the remote **backbone-dest-mac** must be explicitly configured on the PBB tunnel so that the ingress PBB node can build the 802.1ah encapsulation.

If the configured remote backbone-destination-mac is not known in the local FDB, the Epipe customer frames will be 802.1ah encapsulated and flooded into the B-VPLS until the MAC is learned. As previously discussed, MMRP does not help to minimize the flooding because the PBB Epipes always use the configured **backbone-destination-mac** for flooding traffic as opposed to the group B-MAC derived from the ISID.

Flooding could be indefinitely prolonged in the following cases:

- Configuration mistake of the **backbone-destination-mac**. The service will not work but the operator will not detect the mistake since the customer traffic is not dropped at the source node. Every single frame is turned into an unknown unicast PBB frame and hence flooded into the B-VPLS domain.
- Change the **backbone-smac** in the remote PE B-VPLS instance.
- There is only unidirectional traffic in the Epipe service. In this case, the backbone-dest-mac will never be learned in the local SFIB and the frames will always be flooded into the B-VPLS domain.
- The remote node owning the **backbone-destination-mac** simply goes down.

In any of those cases, the operator can easily check whether the PBB Epipe is flooding into the B-VPLS domain, just by looking at the flood flag in the following command output:

```
A:MTU-1# show service id 3 base
=====
Service Basic Information
=====
Service Id      : 3                Vpn Id         : 0
Service Type   : Epipe
Name           : (Not Specified)
Description    : pbb epipe number 3
Customer Id    : 1                Creation Origin : manual
Last Status Change: 01/08/2015 14:39:40
Last Mgmt Change  : 01/08/2015 14:38:48
Test Service   : No
Admin State    : Up                Oper State     : Up
MTU            : 1514
Vc Switching  : False
SAP Count     : 1                SDP Bind Count : 0
Per Svc Hashing : Disabled
Force QTag Fwd : Disabled

-----
Service Access & Destination Points
-----
```

## Flood Avoidance in PBB-Epipes

```
Identifier                               Type           AdmMTU  OprMTU  Adm  Opr
-----
sap:1/1/4:5                             q-tag         1518    1518    Up   Up

-----
PBB Tunnel Point
-----
B-vpls      Backbone-dest-MAC  Isid      AdmMTU  OperState  Flood  Oper-dest-MAC
-----
101         MTU-3              3         2000    Up         Yes    00:31:31:31:31:31
-----
Last Status Change: 01/08/2015 14:39:40
Last Mgmt Change   : 01/08/2015 14:38:48
=====
A:MTU-1#
```

In this particular example, the PBB Epipe 3 is flooding into the B-VPLS 101, as the flood flag indicates. The operator can also confirm that the operational destination B-MAC for the pbb-tunnel, MTU-3, has not been learned in the B-VPLS FDB:

```
A:MTU-1# show service id 101 fdb pbb
=====
Forwarding Database, b-Vpls Service 101
=====
MAC                Source-Identifier  iVplsMACs  Epipes  Type/Age
-----
No Matching Entries
=====
A:MTU-1#
```

## Flooding Cases 1 and 2 — Wrong backbone-dest-mac

Flooding cases 1 and 2 should be fixed after detecting the flooding (see previous commands) and checking the FDBs and PBB tunnel configurations.

## Flooding Case 3 — Unidirectional Traffic: Virtual MEP and CCM Configuration

For flooding case 3 (unidirectional traffic), Alcatel-Lucent recommends the use of ETH-CFM (802.1ag/Y.1731 Connectivity Fault Management) virtual Maintenance End Points (MEPs). By defining a virtual MEP per node terminating a PBB-Epipe, configuring the MEP mac-address to be the source-bmac value and activating continuity check messages (ccm) we achieve a twofold effect:

- The **pbb-tunnel backbone-destination-mac** will always be learned at the local FDB, as long as the remote virtual MEP is active and sending **cc** messages. As a result, there will not be flooding even if we have unidirectional traffic.
- An automatic proactive OAM mechanism exists to detect failures on remote nodes, which ultimately cause unnecessary flooding in the B-VPLS domain.

In the following network example, the virtual MEPs in B-VPLS 101: MEP11, MEP21 and MEP31 are configured.

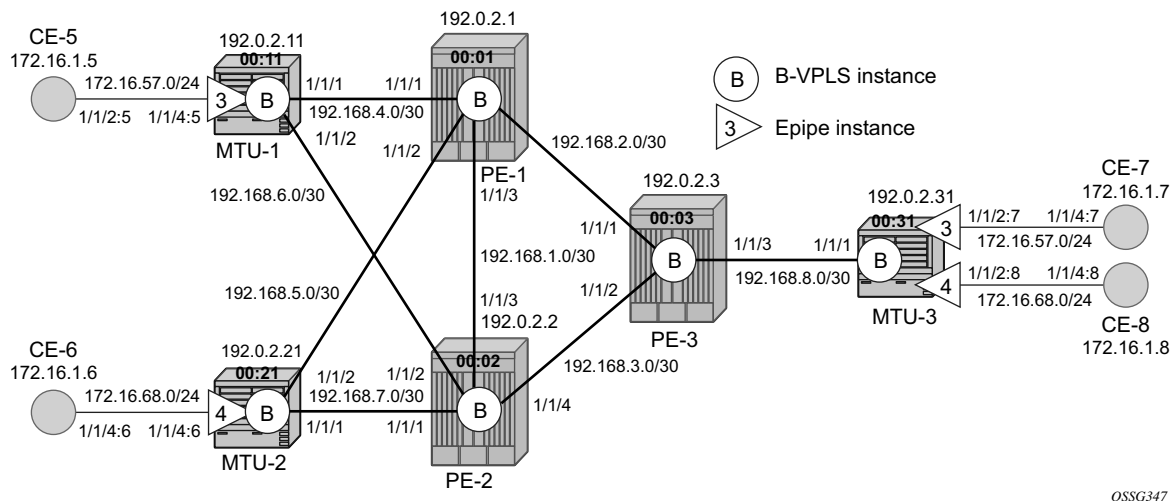


Figure 76: Virtual MEPs for Flooding Avoidance

## Flood Avoidance in PBB-Epipes

The following configuration example uses MTU-1. First, the general ETH-CFM configuration is made:

```
configure
  eth-cfm
    domain 1 format none level 3
      association 1 format icc-based name "B-VPLS-000101"
        bridge-identifier 101
        exit
        remote-mepid 21
        remote-mepid 31
      exit
    exit
  exit
exit
```

Then the actual virtual MEP configuration is made:

```
configure
  service
    vpls 101
      eth-cfm
        mep 11 domain 1 association 1
          ccm-enable
          mac-address 00:11:11:11:11:11
          no shutdown
        exit
      exit
    exit
  exit
```

Note that the MAC address configured for the MEP11 matches the MAC address configured as the **source-bmac** on MTU-1, which is the **backbone-destination-mac** configured on the Epipe 3 pbb-tunnel on MTU-3:

```
# for MTU-1
configure
  service
    pbb
      source-bmac 00:11:11:11:11:11
      mac-name "MTU-1" 00:11:11:11:11:11
      mac-name "MTU-2" 00:21:21:21:21:21
      mac-name "MTU-3" 00:31:31:31:31:31
    exit
```

```
# for MTU-3
configure
  service
    pbb
      source-bmac 00:31:31:31:31:31
      mac-name "MTU-1" 00:11:11:11:11:11
      mac-name "MTU-2" 00:21:21:21:21:21
      mac-name "MTU-3" 00:31:31:31:31:31
    exit
  epipe 3 customer 1 create
    description "pbb epipe number 3"
```



```

pbb
  tunnel 101 backbone-dest-mac "MTU-1" isid 3
exit
sap 1/1/2:7 create
exit
no shutdown
exit

```

Once MEP11 has been configured, check that MTU-3 is receiving **cc** messages from MEP11 with the following command:

```

*A:MTU-3# show eth-cfm mep 31 domain 1 association 1 all-remote-mepids
=====
Eth-CFM Remote-Mep Table
=====
R-mepId AD Rx CC RxRdi Port-Tlv If-Tlv Peer Mac Addr CCM status since
-----
11      True False Absent Absent 00:11:11:11:11:11 01/08/2015 15:07:07
21      True False Absent Absent 00:21:21:21:21:21 01/08/2015 15:07:07
=====
Entries marked with a 'T' under the 'AD' column have been auto-discovered.
*A:MTU-3#

```

As a result of the **cc** messages coming from MEP11, the MTU-1 MAC is permanently learned in the B-VPLS 101 FDB on node MTU-3, and no flooding exists:

```

*A:MTU-3# show service id 101 fdb pbb
=====
Forwarding Database, b-Vpls Service 101
=====
MAC                Source-Identifier    iVplsMACs  Epipes  Type/Age
-----
00:11:11:11:11:11 sdp:33:101          0           1       L/0
00:21:21:21:21:21 sdp:33:101          0           1       L/0
ea:4b:ff:00:00:00 sdp:33:101          0           0       L/0
=====
*A:MTU-3#

*A:MTU-3# show service id 3 base
=====
Service Basic Information
=====
Service Id          : 3                Vpn Id           : 0
Service Type        : Epipe
Name                 : (Not Specified)
Description          : pbb epipe number 3
Customer Id         : 1                Creation Origin   : manual
Last Status Change : 01/08/2015 14:42:37
Last Mgmt Change    : 01/08/2015 14:41:33
Test Service        : No
Admin State         : Up                Oper State        : Up
MTU                  : 1514
Vc Switching        : False
SAP Count           : 1                SDP Bind Count    : 0
Per Svc Hashing     : Disabled
Force QTag Fwd      : Disabled

```

## Flood Avoidance in PBB-Epipes

```
-----  
Service Access & Destination Points  
-----  
Identifier                               Type           AdmMTU  OprMTU  Adm  Opr  
-----  
sap:1/1/2:7                             q-tag         1518    1518    Up   Up  
-----  
PBB Tunnel Point  
-----  
B-vpls      Backbone-dest-MAC  Isid      AdmMTU  OperState  Flood  Oper-dest-MAC  
-----  
101         MTU-1              3         2000    Up         No      00:11:11:11:11:11  
-----  
Last Status Change: 01/08/2015 14:42:37  
Last Mgmt Change   : 01/08/2015 14:41:33  
=====
```

\*A:MTU-3#

## Flooding Case 4 — Remote Node Failure

If the node owner of the **backbone-dest-mac** fails or gets isolated, the node where the PBB Epipe is initiated will not detect the failure; that is, if MTU-1 fails, the Epipe 3 remote end will also fail but MTU-3 will not detect the failure and as a result of that, MTU-3 will flood the traffic to the network (flooding will occur after MTU-1 MAC is removed from the B-VPLS FDBs, due to either the B-VPLS flushing mechanisms or aging).

In order to avoid/reduce flooding in this case, the following mechanisms are recommended:

- Provision virtual MEPs in the B-VPLS instances terminating PBB Epipes, as already explained. This will guarantee there is no unknown B-MAC unicast being flooded under normal operation.
- CCM timers should be provisioned based on how long the service provider is willing to accept flooding.

```
*A:MTU-3# configure eth-cfm domain 1 association 1 ccm-interval
- ccm-interval {interval}
- no ccm-interval
<interval> : {1|10|60|600} - default 10 seconds
```

- From 8.0R1 onward, it is possible to provision discard-unknown in the B-VPLS on the MTUs, i.e. MTU-1, MTU-2 and MTU-3, so that flooded traffic due to the destination MAC being unknown in the B-VPLS is discarded immediately at the MTU. Note that it is important to configure this in conjunction with the CC messages from the virtual MEPs to ensure that the remote B-MACs are learned in both directions. If for any reason the remote B-MACs are not in the MTU B-VPLS, no traffic will be forwarded at all on the PBB-Epipe.
- As soon as the MTU node recovers, it will start sending CC messages and the backbone-mac will be learned on the backbone and MTU nodes again.

```
*A:PE-1# configure service vpls 101 discard-unknown
*A:PE-2# configure service vpls 101 discard-unknown
*A:PE-3# configure service vpls 101 discard-unknown
```

With the recommended configuration in place, in case MTU-1 fails, the **backbone-dest-mac** configured on the pbb-tunnel for Epipe 3 on MTU-3 will be removed from the B-VPLS 101 on all the nodes (either by MAC flush mechanisms on the B-VPLS or by aging). From that point on, traffic originated from CE-7 will be discarded at MTU-3 and won't be flooded further.

As soon as MTU-1 comes back up, MEP11 will start sending CCM and as such the MTU-1 MAC will be learned throughout the B-VPLS 101 domain and in particular in PE-1, PE-3 and MTU-3 (note that CCM PDUs use a multicast address). From the moment MTU-1 MAC is known on the backbone nodes and MTU-3, the traffic won't be discarded any more, but forwarded to MTU-1.

## PBB-Epipe Show Commands

The following commands can help to check the PBB Epipe configuration and their related parameters.

For the B-VPLS service:

```
*A:MTU-1# show service id 101 base
=====
Service Basic Information
=====
Service Id       : 101                Vpn Id           : 0
Service Type    : b-VPLS
Name             : (Not Specified)
Description      : (Not Specified)
Customer Id     : 1                  Creation Origin   : manual
Last Status Change: 01/08/2015 14:39:40
Last Mgmt Change : 01/08/2015 14:38:48
Etree Mode      : Disabled
Admin State     : Up                 Oper State        : Up
MTU             : 2000              Def. Mesh VC Id  : 101
SAP Count       : 0                 SDP Bind Count   : 2
Snd Flush on Fail : Disabled        Host Conn Verify : Disabled
Propagate MacFlush: Disabled        Per Svc Hashing  : Disabled
Allow IP Intf Bind: Disabled
Temp Flood Time : Disabled          Temp Flood        : Inactive
Temp Flood Chg Cnt: 0
VSD Domain      : <none>
Oper Backbone Src : 00:11:11:11:11:11
Use SAP B-MAC   : Disabled
i-Vpls Count    : 0
Epipe Count    : 1
=====
Service Access & Destination Points
=====
Identifier                               Type      AdmMTU  OprMTU  Adm  Opr
-----
sdp:111:101 S(192.0.2.1)                 Spok      8000    8000    Up   Up
sdp:112:101 S(192.0.2.2)                 Spok      8000    8000    Up   Up
=====
*A:MTU-1#
```

For the Epipe service:

```
*A:MTU-1# show service id 3 base
=====
Service Basic Information
=====
Service Id       : 3                Vpn Id           : 0
Service Type    : Epipe
Name             : (Not Specified)
Description      : pbb epipe number 3
Customer Id     : 1                  Creation Origin   : manual
Last Status Change: 01/08/2015 14:39:40
Last Mgmt Change : 01/08/2015 14:38:48
```

```

Test Service      : No
Admin State       : Up           Oper State        : Up
MTU               : 1514
Vc Switching     : False
SAP Count        : 1           SDP Bind Count   : 0
Per Svc Hashing  : Disabled
Force QTag Fwd   : Disabled

```

```
-----
Service Access & Destination Points
-----
```

Identifier	Type	AdmMTU	OprMTU	Adm	Opr
sap:1/1/4:5	q-tag	1518	1518	Up	Up

```
-----
PBB Tunnel Point
-----
```

B-vpls	Backbone-dest-MAC	Isid	AdmMTU	OperState	Flood	Oper-dest-MAC
101	MTU-3	3	2000	Up	No	00:31:31:31:31:31

```
Last Status Change: 01/08/2015 14:39:40
```

```
Last Mgmt Change : 01/08/2015 14:38:48
```

```
=====
*A:MTU-1#
```

The following command shows all the Epipe instances multiplexed into a particular B-VPLS and its status.

```
*A:MTU-1# show service id 101 epipe
```

```
-----
Related Epipe services for b-Vpls service 101
-----
```

Epipe SvcId	Oper	ISID	Admin	Oper
3	3	Up	Up	

```
-----
Number of Entries : 1
-----
=====
```

To check the virtual MEP information, show the local virtual MEPs configured on the node:

```
* A:MTU-1# show eth-cfm cfm-stack-table all-virtuals
```

```
-----
CFM Stack Table Defect Legend:
```

```
R = Rdi, M = MacStatus, C = RemoteCCM, E = ErrorCCM, X = XconCCM
```

```
A = AisRx, L = CSF LOS Rx, F = CSF AIS/FDI rx, r = CSF RDI rx
```

```
-----
CFM Virtual Stack Table
-----
```

Service	Lvl	Dir	Md-index	Ma-index	MepId	Mac-address	Defect
101	3	U	1	1	11	00:11:11:11:11:11	-----

```
-----
*A:MTU-1#
```

## PBB-Epipe Show Commands

The following command shows all the information related to the remote MEPs configured in the association, for example, the remote virtual MEPs configured in MTU-2 and MTU-3:

```
*A:MTU-1# show eth-cfm mep 11 domain 1 association 1 all-remote-mepids
=====
Eth-CFM Remote-Mep Table
=====
R-mepId AD Rx CC RxRdi Port-Tlv If-Tlv Peer Mac Addr      CCM status since
-----
21          True  False Absent  Absent 00:21:21:21:21:21 01/08/2015 14:58:12
31          True  False Absent  Absent 00:31:31:31:31:31 01/08/2015 15:04:24
=====
Entries marked with a 'T' under the 'AD' column have been auto-discovered.
*A:MTU-1#
```

The following command shows the detail information and status of the local virtual MEP configured in MTU-1:

```
*A:MTU-1# show eth-cfm mep 11 domain 1 association 1
=====
Eth-Cfm MEP Configuration Information
=====
Md-index      : 1                      Direction     : Up
Ma-index      : 1                      Admin        : Enabled
MepId        : 11                     CCM-Enable   : Enabled
SvcId        : 101
Description   : (Not Specified)
FngAlarmTime : 0                      FngResetTime : 0
FngState     : fngReset                ControlMep   : False
LowestDefectPri : macRemErrXcon        HighestDefect : none
Defect Flags  : None
Mac Address   : 00:11:11:11:11:11
CcmLtmPriority : 7                      CcmPaddingSize : 0 octets
CcmTx        : 60                      CcmSequenceErr : 0
CcmIgnoreTLVs : (Not Specified)
Fault Propagation: disabled             FacilityFault : n/a
MA-CcmInterval : 10                    MA-CcmHoldTime : 0ms
MA-Primary-Vid : Disabled
Eth-1Dm Threshold: 3(sec)              MD-Level     : 3
Eth-Ais      : Disabled
Eth-Ais Tx defCCM: allDef
Eth-Tst     : Disabled
Eth-CSF     : Disabled

Redundancy:
  MC-LAG State : n/a

CcmLastFailure Frame:
  None

XconCcmFailure Frame:
  None
=====
*A:MTU-1#
```

When there is a failure on a remote Epipe node, as discussed, the source node keeps sending traffic. The 802.1ag/Y.1731 virtual MEP configured can help to detect and troubleshoot the problem. For instance, when a failure happens in MTU-3 (node goes down or the B-VPLS instance is shut down), the virtual MEP show commands will show the following information:

```
*A:MTU-1# show eth-cfm mep 11 domain 1 association 1
=====
Eth-Cfm MEP Configuration Information
=====
Md-index          : 1                      Direction       : Up
Ma-index          : 1                      Admin           : Enabled
MepId             : 11                     CCM-Enable     : Enabled
SvcId             : 101
Description       : (Not Specified)
FngAlarmTime     : 0                      FngResetTime   : 0
FngState         : fngDefectReported      ControlMep     : False
LowestDefectPri  : macRemErrXcon          HighestDefect  : defRemoteCCM
Defect Flags     : bDefRDICCM bDefRemoteCCM
Mac Address      : 00:11:11:11:11:11
CcmLtmPriority   : 7                      CcmPaddingSize : 0 octets
CcmTx            : 66                     CcmSequenceErr : 0
CcmIgnoreTLVs   : (Not Specified)
Fault Propagation: disabled                FacilityFault  : n/a
MA-CcmInterval  : 10                     MA-CcmHoldTime : 0ms
MA-Primary-Vid  : Disabled
Eth-1Dm Threshold: 3(sec)                 MD-Level       : 3
Eth-Ais         : Disabled
Eth-Ais Tx defCCM: allDef
Eth-Tst         : Disabled
Eth-CSF         : Disabled

Redundancy:
  MC-LAG State : n/a

CcmLastFailure Frame:
  None

XconCcmFailure Frame:
  None
=====
*A:MTU-1#
```

The bDefRemoteCCMdefect flag clearly shows that there is a remote MEP in the association which has stopped sending CCMs. In order to find out which node is affected, see the following output:

```
*A:MTU-1# show eth-cfm mep 11 domain 1 association 1 all-remote-mepids
=====
Eth-CFM Remote-Mep Table
=====
R-mepId AD Rx CC RxRdi Port-Tlv If-Tlv Peer Mac Addr      CCM status since
-----
21      True True Absent Absent 00:21:21:21:21:21 01/08/2015 14:58:12
31      False False Absent Absent 00:00:00:00:00:00 01/08/2015 15:09:00
=====
Entries marked with a 'T' under the 'AD' column have been auto-discovered.
```

## PBB-Epipe Show Commands

\*A:MTU-1#

CCMs are no longer received from virtual MEP 31 (the one defined in MTU-3) and since 12/02/2009 19:47:37. This conveys which node has failed and when.



## Conclusion

Point-to-Point Ethernet services can use the same operational model followed by PBB VPLS for multipoint services. In other words, Epipes can be linked to the same B-VPLS domain being used by I-VPLS instances and use the existing H-VPLS network infrastructure in the core. The use of PBB Epipes reduces dramatically the number of services and pseudowires in the core and therefore allows the service provider to scale the number of ELINE services in the network.

The example used in this document shows the configuration of the PBB Epipes as well as all the related features which are required for this environment. Show commands have also been suggested so that the operator can verify and troubleshoot the service.

Conclusion