CONTENTS

MPLS L2 VPN - VPLS - Multipoint Service

« MPLS L2 VPN - VLAN Based Point-to-Point EoMPLS Service (/workbook/view/service-provider-v4/task/mpls-l2-vpn-vlan-based-point-to-point-eompls-service-Mjk0NA%3D%3D) | MPLS L2 VPN - H-VPLS - Multipoint Service (/workbook/view/service-provider-v4/task/mpls-l2-vpn-h-vpls-multipoint-service-Mjk0OA%3D%3D) »

Last updated: April 22, 2016

Note:

Initial Configuration & Diagrams: Load the initial configuration files for the section named **L2VPN**, which can be found in CCIE SPv4 Topology Diagrams & Initial Configurations (http://labs.ine.com/workbook/view/service-provider-v4/task/ccie-spv4-topology-diagrams-initial-configs). Refer to the **L2VPN Port Based Diagram** in order to complete this task.

A Note L2VPN Platform Support: L2VPN data plane forwarding is not currently supported in XRv, however the control plan functionality is. Throughout the L2VPN sections of the workbook, we will configure the different variations of L2VPN on the XRv platform to explore IOS-XR specific syntax, and to validate the protocol mechanics via the control-plane state.

Task

- Configure R2, R4, R5 and XR1 for a Multipoint L2VPN over MPLS as follows:
  - The attachment circuits are the Ethernet links connecting to R1, R7, R8 and XR2 respectively.
  - R2, R4, R5, and XR2 should provide connectivity between R1, R7, R8, and XR2's VLAN 100.
  - The configuration should VLAN 100 between R1, R7, R8, and XR2 to be extended over the MPLS network.
- Once complete, R1, R7 and R8 should form an OSPFv2 adjacencies and have IP reachability to each other's Looopback0 networks.
- Note that XR2 will not form an OSPFv2 adjacency due to the XRv L2VPN limitation.

Configuration **Click to *collapse***

```
!


R2:
interface GigabitEthernet2
 no ip address
 speed 1000
 no negotiation auto
 service instance 100 ethernet
  encapsulation dot1q 100
  bridge-domain 100
 !
l2 vfi VPLS_R1_R7_R8_XR2 manual
 vpn id 100
 bridge-domain 100
 neighbor 5.5.5.5 encapsulation mpls
 neighbor 4.4.4.4 encapsulation mpls
 neighbor 19.19.19.19 encapsulation mpls



R4:
interface GigabitEthernet2
 no ip address
 speed 1000
 no negotiation auto
 service instance 100 ethernet
  encapsulation dot1q 100
  bridge-domain 100
 !
l2 vfi VPLS_R1_R7_R8_XR2 manual
 vpn id 100
 bridge-domain 100
 neighbor 2.2.2.2 encapsulation mpls
 neighbor 5.5.5.5 encapsulation mpls
 neighbor 19.19.19.19 encapsulation mpls



R5:
interface GigabitEthernet2
 no ip address
 speed 1000
 no negotiation auto
 service instance 100 ethernet
  encapsulation dot1q 100
  bridge-domain 100
 !
l2 vfi VPLS_R1_R7_R8_XR2 manual
 vpn id 100
 bridge-domain 100
 neighbor 2.2.2.2 encapsulation mpls
 neighbor 4.4.4.4 encapsulation mpls
 neighbor 19.19.19.19 encapsulation mpls
```

```
XR1:
interface GigabitEthernet0/0/0/1.100 l2transport
 dot1q vlan 100
!
l2vpn
 pw-class VPLS_CLASS
  encapsulation mpls
  !
 !
 bridge group VPLS
  bridge-domain 100
   interface GigabitEthernet0/0/0/1.100
   !
   vfi 100
    neighbor 2.2.2.2 pw-id 100 pw-class VPLS_CLASS
    !
    neighbor 4.4.4.4 pw-id 100 pw-class VPLS_CLASS
    !
    neighbor 5.5.5.5 pw-id 100 pw-class VPLS_CLASS
    !
   !
  !
 !
!
```

```
!
!
! Alternate IOS Config - Bridge Domain Centric
!
!
```

```
R2:
interface GigabitEthernet2
 no ip address
 negotiation auto
 service instance 100 ethernet
  encapsulation dot1q 100
!
l2vpn vfi context VPLS_R1_R7_R8_XR2
 vpn id 100
 member 19.19.19.19 encapsulation mpls
 member 5.5.5.5 encapsulation mpls
 member 4.4.4.4 encapsulation mpls
!
bridge-domain 100
 member GigabitEthernet2 service-instance 100
 member vfi VPLS_R1_R7_R8_XR2
```

```
R4:
```

```
interface GigabitEthernet2
 no ip address
 negotiation auto
 service instance 100 ethernet
  encapsulation dot1q 100
!
l2vpn vfi context VPLS_R1_R7_R8_XR2
 vpn id 100
 member 19.19.19.19 encapsulation mpls
 member 5.5.5.5 encapsulation mpls
 member 2.2.2.2 encapsulation mpls
!
bridge-domain 100
 member GigabitEthernet2 service-instance 100
 member vfi VPLS_R1_R7_R8_XR2



R5:
interface GigabitEthernet2
 no ip address
 negotiation auto
 service instance 100 ethernet
  encapsulation dot1q 100
!
l2vpn vfi context VPLS_R1_R7_R8_XR2
 vpn id 100
 member 19.19.19.19 encapsulation mpls
 member 4.4.4.4 encapsulation mpls
 member 2.2.2.2 encapsulation mpls
!
bridge-domain 100
 member GigabitEthernet2 service-instance 100
 member vfi VPLS_R1_R7_R8_XR2
```

Verification

Virtual Private LAN Services (VPLS) is similar to EoMPLS, with the exception that VPLS emulates a multipoint topology, while EoMPLS only emulates point-to-point topologies.

Instead of using a single pseudowire between two PEs to provide a point-to-point connection between two Access Circuits, VPLS uses a full mesh of pseudowires between PEs to provide an emulated LAN between all Access Circuits participating in the service. Each PE builds a pseudowire to all other PEs where a client exist. VPLS uses a Virtual Forwarding Instance (VFI) to host all of the pseudowires for a particular service. PEs can have more than one VFI - essentially one per VPLS instance they are servicing. Note that some Cisco platforms refer to the VFI as a VSI, so these terms may be used interchangeably in some documentation.

A major difference between VPLS and EoMPLS is the data-plane driven MAC learning that each PE participates in. In a point-to-point EoMPLS service, the PE has a single pseudowire to which it can forward the Access Circuit frames. However, in a VPLS service, the PE has multiple pseudowires which can be used to forward the frame. In order to cope with this issue, the PE performs MAC learning on the VFI and maps the source MAC to receiving pseudowire in the forwarding table.

Note that two separate config samples were provided in the solution. Although the result is the same, the structure of the configuration is a bit different. Both clearly illustrate the forwarding model employed by VPLS.

- Frames coming in with outer VLAN tag 100 will be mapped to the EFP being referenced by Service Instance 100.

```
interface GigabitEthernet2
 no ip address
 speed 1000
 no negotiation auto
 service instance 100 ethernet
  encapsulation dot1q 100
```

- The forwarding action being taken for this EFP is to send the unmodified frame (no VLAN acrobatics performed) into Bride-Domain 100.

```
  bridge-domain 100
 !
```

- The VFI is instantiated, referencing Bridge-Domain 100.

```
l2 vfi VPLS_R1_R7_R8_XR2 manual
 vpn id 100
 bridge-domain 100
```

- All of the pseudowires which can be used for forwarding in the VFI are defined (Full Mesh between the PEs).

```
 neighbor 5.5.5.5 encapsulation mpls
 neighbor 4.4.4.4 encapsulation mpls
 neighbor 19.19.19.19 encapsulation mpls
```

Validation of the functionality yielded by this multipoint service can be done from the clients point of view. R1, R7, and R8 have formed a full mesh of OSPF adjacencies. It is as if they were connected to the same LAN segment, which validates that we have properly configured a multipoint service over MPLS (VPLS). Note that XR1 is not reachable over the service due to a limitation in the XRv implementation of L2VPN data-plane forwarding.

```
R1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
7.7.7.7           1   FULL/BDR        00:00:39    10.0.100.7      GigabitEthernet2.100
8.8.8.8           1   FULL/DR         00:00:30    10.0.100.8      GigabitEthernet2.100

R7#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           1   FULL/DROTHER    00:00:38    10.0.100.1      GigabitEthernet2.100
8.8.8.8           1   FULL/DR         00:00:37    10.0.100.8      GigabitEthernet2.100

R8#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           1   FULL/DROTHER    00:00:31    10.0.100.1      GigabitEthernet2.100
7.7.7.7           1   FULL/BDR        00:00:38    10.0.100.7      GigabitEthernet2.100

R8#ping 255.255.255.255
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:

 Reply to request 0 from 10.0.100.7, 4 ms
 Reply to request 0 from 10.0.100.1, 4 ms
Reply to request 1 from 10.0.100.7, 4 ms
Reply to request 1 from 10.0.100.1, 4 ms
Reply to request 2 from 10.0.100.7, 5 ms
Reply to request 2 from 10.0.100.1, 5 ms
Reply to request 3 from 10.0.100.7, 4 ms
Reply to request 3 from 10.0.100.1, 4 ms
Reply to request 4 from 10.0.100.7, 6 ms
Reply to request 4 from 10.0.100.1, 6 ms
```

The pseudowires being used here are no different from the ones seen in previous sections. These pseudowires are leveraging a Targeted LDP session in order to exchange the "VC Label". Although XR1 is not able to perform data-plane forwarding, the control-plane portion is fully functional and can be validated. Notice that the labels have been properly exchanged and the mesh of pseudowires are up.

```
RP/0/0/CPU0:XR1#show l2vpn bridge-domain detail
Sat Jun 27 01:14:05.602 UTC
Legend: pp = Partially Programmed.
 Bridge group: VPLS, bridge-domain: 100, id: 0, state: up , ShgId: 0, MSTi: 0
  Coupled state: disabled
  MAC learning: enabled
  MAC withdraw: enabled
    MAC withdraw for Access PW: enabled
    MAC withdraw sent on: bridge port up
    MAC withdraw relaying (access to access): disabled
  Flooding:
    Broadcast & Multicast: enabled
    Unknown unicast: enabled
  MAC aging time: 300 s, Type: inactivity
  MAC limit: 4000, Action: none, Notification: syslog
  MAC limit reached: no
  MAC port down flush: enabled
  MAC Secure: disabled, Logging: disabled
  Split Horizon Group: none
  Dynamic ARP Inspection: disabled, Logging: disabled
  IP Source Guard: disabled, Logging: disabled
  DHCPv4 snooping: disabled
  IGMP Snooping: enabled
  IGMP Snooping profile: none
  MLD Snooping profile: none
  Storm Control: disabled
  Bridge MTU: 1500
  MIB cvplsConfigIndex: 1
  Filter MAC addresses:
  P2MP PW: disabled
  Create time: 26/06/2015 23:50:15 (01:23:50 ago)
  No status change since creation
  ACs: 1 (1 up), VFIs: 1, PWs: 3 (3 up), PBBs: 0 (0 up)
  List of ACs:
    AC: GigabitEthernet0/0/0/1.100, state is up
      Type VLAN; Num Ranges: 1
       VLAN ranges: [100, 100]
      MTU 1500; XC ID 0x1; interworking none
      MAC learning: enabled
      Flooding:
        Broadcast & Multicast: enabled
        Unknown unicast: enabled
      MAC aging time: 300 s, Type: inactivity
      MAC limit: 4000, Action: none, Notification: syslog
      MAC limit reached: no
      MAC port down flush: enabled
      MAC Secure: disabled, Logging: disabled
      Split Horizon Group: none
      Dynamic ARP Inspection: disabled, Logging: disabled
      IP Source Guard: disabled, Logging: disabled
      DHCPv4 snooping: disabled
      IGMP Snooping: enabled
      IGMP Snooping profile: none
```

```
    MLD Snooping profile: none
    Storm Control: disabled
    Static MAC addresses:
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
    Storm control drop counters:
      packets: broadcast 0, multicast 0, unknown unicast 0
      bytes: broadcast 0, multicast 0, unknown unicast 0
    Dynamic ARP inspection drop counters:
      packets: 0, bytes: 0
    IP source guard drop counters:
      packets: 0, bytes: 0
List of Access PWs:
List of VFIs:
  VFI 100 (up)
    PW: neighbor 2.2.2.2, PW ID 100, state is up ( established )
      PW class not set, XC ID 0xff000002
      Encapsulation MPLS, protocol LDP
      Source address 19.19.19.19
      PW type Ethernet, control word disabled, interworking none
      Sequencing not set

      PW Status TLV in use
        MPLS         Local                          Remote
        ------------ ------------------------------ ------------------------
        Label        16006                          31
        Group ID     0x0                            0x0
        Interface    100                            unknown
        MTU          1500                           1500
        Control word disabled                       disabled
        PW type      Ethernet                       Ethernet
        VCCV CV type 0x2                            0x2
                     (LSP ping verification)        (LSP ping verification)
        VCCV CC type 0x6                            0x6
                     (router alert label)           (router alert label)
                     (TTL expiry)                   (TTL expiry)
        ------------ ------------------------------ ------------------------
      Incoming Status (PW Status TLV):
        Status code: 0x0 (Up) in Notification message
      MIB cpwVcIndex: 4278190082
      Create time: 26/06/2015 23:50:15 (01:23:50 ago)
      Last time status changed: 27/06/2015 00:46:07 (00:27:57 ago)
      Last time PW went down: 27/06/2015 00:42:48 (00:31:17 ago)
      MAC withdraw messages: sent 0, received 0
      Static MAC addresses:
      Statistics:
        packets: received 0, sent 0
        bytes: received 0, sent 0
      Storm control drop counters:
        packets: broadcast 0, multicast 0, unknown unicast 0
        bytes: broadcast 0, multicast 0, unknown unicast 0
    DHCPv4 snooping: disabled
    IGMP Snooping profile: none
```

```
MLD Snooping profile: none
 PW: neighbor 4.4.4.4, PW ID 100, state is up ( established )
  PW class not set, XC ID 0xff000003
  Encapsulation MPLS, protocol LDP
  Source address 19.19.19.19
  PW type Ethernet, control word disabled, interworking none
  Sequencing not set

  PW Status TLV in use
    MPLS         Local                          Remote
    ------------ ------------------------------ -------------------------
     Label        16007                          26
    Group ID     0x0                            0x0
    Interface    100                            unknown
    MTU          1500                           1500
    Control word disabled                       disabled
    PW type      Ethernet                       Ethernet
    VCCV CV type 0x2                            0x2
                 (LSP ping verification)        (LSP ping verification)
    VCCV CC type 0x6                            0x6
                 (router alert label)           (router alert label)
                 (TTL expiry)                   (TTL expiry)
    ------------ ------------------------------ -------------------------
  Incoming Status (PW Status TLV):
    Status code: 0x0 (Up) in Notification message
  MIB cpwVcIndex: 4278190083
  Create time: 26/06/2015 23:50:15 (01:23:50 ago)
  Last time status changed: 27/06/2015 00:47:51 (00:26:14 ago)
  Last time PW went down: 27/06/2015 00:42:55 (00:31:10 ago)
  MAC withdraw messages: sent 0, received 0
  Static MAC addresses:
  Statistics:
    packets: received 0, sent 0
    bytes: received 0, sent 0
  Storm control drop counters:
    packets: broadcast 0, multicast 0, unknown unicast 0
    bytes: broadcast 0, multicast 0, unknown unicast 0
DHCPv4 snooping: disabled
IGMP Snooping profile: none
MLD Snooping profile: none
 PW: neighbor 5.5.5.5, PW ID 100, state is up ( established )
  PW class not set, XC ID 0xff000004
  Encapsulation MPLS, protocol LDP
  Source address 19.19.19.19
  PW type Ethernet, control word disabled, interworking none
  Sequencing not set

  PW Status TLV in use
    MPLS         Local                          Remote
    ------------ ------------------------------ -------------------------
     Label        16009                          21
    Group ID     0x0                            0x0
    Interface    100                            unknown
    MTU          1500                           1500
```

```
          Control word disabled                       disabled
          PW type      Ethernet                       Ethernet
          VCCV CV type 0x2                            0x2
                       (LSP ping verification)        (LSP ping verification)
          VCCV CC type 0x6                            0x6
                       (router alert label)           (router alert label)
                       (TTL expiry)                   (TTL expiry)
          ------------ ------------------------------ -------------------------
        Incoming Status (PW Status TLV):
          Status code: 0x0 (Up) in Notification message
        MIB cpwVcIndex: 4278190084
        Create time: 26/06/2015 23:50:15 (01:23:50 ago)
        Last time status changed: 27/06/2015 00:47:49 (00:26:16 ago)
        Last time PW went down: 27/06/2015 00:42:58 (00:31:06 ago)
        MAC withdraw messages: sent 0, received 0
        Static MAC addresses:
        Statistics:
          packets: received 0, sent 0
          bytes: received 0, sent 0
        Storm control drop counters:
          packets: broadcast 0, multicast 0, unknown unicast 0
          bytes: broadcast 0, multicast 0, unknown unicast 0
      DHCPv4 snooping: disabled
      IGMP Snooping profile: none
      MLD Snooping profile: none
      VFI Statistics:
        drops: illegal VLAN 0, illegal length 0


R2#show l2vpn service all detail
Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority
        UP=Up       DN=Down            AD=Admin Down      IA=Inactive
        SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware
        m=manually selected

  Interface          Group       Encapsulation                   Prio  St  XC St
  ---------          -----       -------------                   ----  --  -----
VPLS name: VPLS_R1_R7_R8_XR2, State: UP
  pw100024                       VPLS_R1_R7_R8_XR2(VFI)          0     UP  UP
  pw100027           core_pw     19.19.19.19:100(MPLS)           0     UP  UP
                                 Local VC label 31
                                 Remote VC label 16006

  pw100023           core_pw     5.5.5.5:100(MPLS)               0     UP  UP
                                 Local VC label 28
                                 Remote VC label 61

  pw100022           core_pw     4.4.4.4:100(MPLS)               0     UP  UP
                                 Local VC label 34
                                 Remote VC label 19

BD name: 100, State: --
  -                              100(BD)                         0     UP  --
  -                              VPLS_R1_R7_R8_XR2(VFI)          0     UP  UP
```

Using a less verbose output on IOS, we can see that virtual interfaces are created for each pseudowire. LDP was used to do the signaling (exchanging labels, and bringing up the pseudowire). Notice that all of these pseudowires are marked with Split-Horizon as Enabled.

```
R2#show l2vpn vfi

Legend: RT=Route-target, S=Split-horizon , Y=Yes , N=No


VFI name: VPLS_R1_R7_R8_XR2, state: up, type: multipoint, signaling: LDP
  VPN ID: 100
  Bridge-Domain 100 attachment circuits:
  Pseudo-port interface: pseudowire100024
  Interface          Peer Address     VC ID        S
   pseudowire100027   19.19.19.19      100          Y
   pseudowire100023   5.5.5.5          100          Y
   pseudowire100022   4.4.4.4          100          Y
```

A full mesh of pseudowires is required in order for the service to work properly. VPLS uses split-horizon in order to prevent loops, which means that a packet cannot come into a pseudowire and be sent out of another pseudowire. For example, for R1 to send frames to R7, R1 needs to have a pseudowire established to the PE servicing R7 (R4 in this case). Although R2 may have pseudowires up to R5 and XR1, which are all part of the same VFI and have learned the MAC addresses for the hosts attached to the VFI (so they could technically forward the frame to the correct PE), the split-horizon rule prevents this from occurring. That is, R5 will receive frames on its pseudowire, and will detect that the destination which in this case is R7, is reachable out of the pseudowire it has to R4. This packet will be dropped, as forwarding it out of the pseudowire would break the Split Horizon rule. As there is no Spanning-Tree running between the pseudowires, a loop can easily be formed without Split Horizon.

Note that Split-Horizon only affects Core Facing pseudowires. All the pseudowires in our network currently are considered "core-pw" as can be seen by the outputs above. The Split-Horizon rules change when the notion of a Spoke pseudowire is introduced in H-VPLS. In this design, traffic coming into a spoke pseudowire can be forwarded out of a core pseudowire.

All MAC learning in VPLS is data plane driven, and behaves like a standard bridge - its dynamic and based on source MAC and VLAN. By default MACs are aged out of a VFI after 5 min of inactivity and will have to be re-learned.

Note the MAC address for R7 and R8 that R1 sees in its ARP cache.

```
R1#show ip arp Gig2.100
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.0.100.1              -   0050.569e.35b6  ARPA   GigabitEthernet2.100
Internet  10.0.100.7            104   0050.569e.0b8a  ARPA   GigabitEthernet2.100
Internet  10.0.100.8            104   0050.569e.30c3  ARPA   GigabitEthernet2.100


The PEs servicing the VFI should have fresh entries in their bridging tables from the OSPF hellos.
```

These same MAC addresses are being learned over their corresponding pseudowire under the VFI. This can be observed by looking at the Bridge Domain.

```
R2#show bridge-domain 100
Bridge-domain 100 (4 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 300 second(s)
    GigabitEthernet2 service instance 100
    vfi VPLS_R1_R7_R8_XR2 neighbor 19.19.19.19 100
    vfi VPLS_R1_R7_R8_XR2 neighbor 4.4.4.4 100
    vfi VPLS_R1_R7_R8_XR2 neighbor 5.5.5.5 100
   AED MAC address    Policy  Tag       Age  Pseudoport
   0   0050.569E.35B6 forward dynamic   300  GigabitEthernet2.EFP100
   0   0050.569E.0B8A forward dynamic   298  VPLS_R1_R7_R8_XR2.1001020
   1   FFFF.FFFF.FFFF flood   static    0    OLIST_PTR:0xe811b870
   0   0050.569E.30C3 forward dynamic   299  VPLS_R1_R7_R8_XR2.1001021
```

Notice that the local MAC address is learned from `GigabitEthernet2.EFP100`, which is EFP 100 on Gig2. This is the virtual interface created for the EFP. The remote MAC addresses are learned over the VFI construct, `VPLS_R1_R7_R8_XR2`.

```
R4#show bridge-domain 100
Bridge-domain 100 (4 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 300 second(s)
    GigabitEthernet2 service instance 100
    vfi VPLS_R1_R7_R8_XR2 neighbor 19.19.19.19 100
    vfi VPLS_R1_R7_R8_XR2 neighbor 5.5.5.5 100
    vfi VPLS_R1_R7_R8_XR2 neighbor 2.2.2.2 100
   AED MAC address    Policy  Tag       Age  Pseudoport
   0   0050.569E.35B6 forward dynamic   293  VPLS_R1_R7_R8_XR2.100101b
   0   0050.569E.0B8A forward dynamic   299  GigabitEthernet2.EFP100
   1   FFFF.FFFF.FFFF flood   static    0    OLIST_PTR:0xe810e000
   0   0050.569E.30C3 forward dynamic   292  VPLS_R1_R7_R8_XR2.100101a

R5#show bridge-domain
Bridge-domain 100 (4 ports in all)
State: UP                    Mac learning: Enabled
Aging-Timer: 300 second(s)
    GigabitEthernet2 service instance 100
    vfi VPLS_R1_R7_R8_XR2 neighbor 19.19.19.19 100
    vfi VPLS_R1_R7_R8_XR2 neighbor 4.4.4.4 100
    vfi VPLS_R1_R7_R8_XR2 neighbor 2.2.2.2 100
   AED MAC address    Policy  Tag       Age  Pseudoport
   0   0050.569E.35B6 forward dynamic   300  VPLS_R1_R7_R8_XR2.100101a
   0   0050.569E.0B8A forward dynamic   296  VPLS_R1_R7_R8_XR2.1001019
   1   FFFF.FFFF.FFFF flood   static    0    OLIST_PTR:0xe80c8800
   0   0050.569E.30C3 forward dynamic   299  GigabitEthernet2.EFP100
```

Notice that there is an entry for FFFF.FFFF.FFFF. This is used to flood multi-destination frames into the bridging domain. If the PE needs to flood a frame (ARP for example), it will use this entry to replicate the frame onto all pseudowires participating in the VFI.

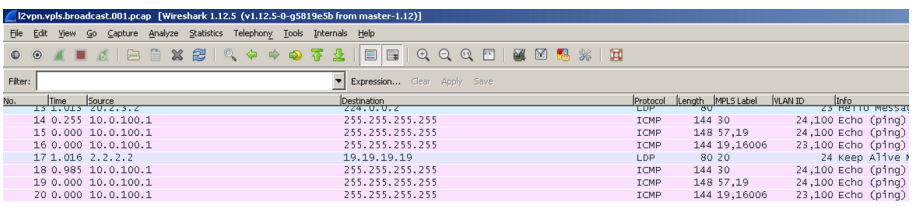
Note that XR1 has not learned any MAC addresses due to the implementation limitation, but the corresponding command to look at the MACs learned over the bride domain is being shown for completeness.

```
RP/0/0/CPU0:XR1#show l2vpn forwarding bridge-domain mac-address location 0/0/CPU0

Sat Jun 27 04:17:01.980 UTC
```

Lets perform a packet capture on R2 as a broadcast frame is replicated out of all pseudowires.

```
R2#monitor capture CAP match any interface Gig1.23 out

R2#monitor capture CAP match any interface Gig1.24 out


R2#monitor capture CAP start

%BUFCAP-6-ENABLE: Capture Point CAP enabled.




R1#clear arp-cache

R1#ping 255.255.255.255

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 255.255.255.255, timeout is 2 seconds:


Reply to request 0 from 10.0.100.7, 5 ms

Reply to request 0 from 10.0.100.8, 5 ms

Reply to request 1 from 10.0.100.7, 5 ms

Reply to request 1 from 10.0.100.8, 6 ms

Reply to request 2 from 10.0.100.7, 5 ms

Reply to request 2 from 10.0.100.8, 5 ms

Reply to request 3 from 10.0.100.7, 5 ms

Reply to request 3 from 10.0.100.8, 5 ms

Reply to request 4 from 10.0.100.7, 5 ms

Reply to request 4 from 10.0.100.8, 5 ms



R2#monitor capture CAP stop

R6#

%BUFCAP-6-DISABLE: Capture Point CAP disabled.


R2#monitor capture CAP export ftp://cisco:cisco@169.254.254.1/l2vpn.vpls.broadcast.001.pcap

Writing l2vpn.vpls.broadcast.001.pcap

Exported Successfully
```

In the packet capture shown below, we can see that R2 takes the broadcast from its Access Circuit to R1 and replicates it out of its three pseudowires to R4, R5, and XR1. Note that two columns were added to the packet capture display for clarity - MPLS Label and VLAN ID. The packets are tagged with Inner VLAN-ID of 100, which is the Customer facing VLAN being extended over VPLS, and outer VLAN ID of 23/24 depending on the pseudowire its being sent on (which core facing interface it is label switched out of)



Packet #14 on the packet trace has a single MPLS Tag of 30 and outer VLAN ID of 24. This is the pseudowire to R4, as R2 and R4 are directly connected over Gig1.24 and the PHP optimization takes place. Packet #15 has two MPLS labels in the stack, 57 and 19. These packets are being encapsulated onto the pseudowire to R5, which is using outer VLAN ID 24, which simply means they are routed out of

Gig1.24, just like the pseudowire to R4. Packet #16 also has two labels in the stack, 19 and 16006, which is the pseudowire to XR1. Notice that these packets are seen with an outer VLAN ID of 23, meaning they are being routed out of Gig1.23.

Notice that the labels seen in the packet capture match the pseudowire labels seen in the following output:

```
R2#show l2vpn service all detail

Legend: St=State    XC St=State in the L2VPN Service      Prio=Priority

        UP=Up       DN=Down             AD=Admin Down      IA=Inactive

        SB=Standby  HS=Hot Standby      RV=Recovering      NH=No Hardware

        m=manually selected


  Interface          Group       Encapsulation                   Prio  St  XC St

  ---------          -----       -------------                   ----  --  -----

VPLS name: VPLS_R1_R7_R8_XR2, State: UP

  pw100031                       VPLS_R1_R7_R8_XR2(VFI)          0     UP  UP

  pw100038           core_pw     19.19.19.19:100(MPLS)           0     UP  UP

                                 Local VC label 17

                                  Remote VC label 16006


  pw100037           core_pw     4.4.4.4:100(MPLS)               0     UP  UP

                                 Local VC label 38

                                  Remote VC label 30


  pw100036           core_pw     5.5.5.5:100(MPLS)               0     UP  UP

                                 Local VC label 16

                                  Remote VC label 19


BD name: 100, State: --

  -                              100(BD)                         0     UP  --

  -                              VPLS_R1_R7_R8_XR2(VFI)          0     UP  UP
```

-

```
R2#show mpls 2transport vc detail | include Destination address|Output interface

  Destination address: 4.4.4.4, VC ID: 100, VC status: up

    Output interface: Gi1.24, imposed label stack {30}

  Destination address: 5.5.5.5, VC ID: 100, VC status: up

    Output interface: Gi1.24, imposed label stack {57 19}

  Destination address: 19.19.19.19, VC ID: 100, VC status: up

    Output interface: Gi1.23, imposed label stack {19 16006}
```

« MPLS L2 VPN - VLAN Based Point-to-Point EoMPLS Service (/workbook/view/service-provider-v4/task/mpls-l2-vpn-vlan-based-point-to-point-eompls-service-Mjk0NA%3D%3D) | MPLS L2 VPN - H-VPLS - Multipoint Service (/workbook/view/service-provider-v4/task/mpls-l2-vpn-h-vpls-multipoint-service-Mjk0OA%3D%3D) »