

MPLS L2 VPN - H-VPLS - Multipoint Service

« [MPLS L2 VPN - VPLS - Multipoint Service \(/workbook/view/service-provider-v4/task/mpls-l2-vpn-vpls-multipoint-service-Mjk0NQ%3D%3D\)](#) | [IP L2 VPN with L2TPv3 \(/workbook/view/service-provider-v4/task/ip-l2-vpn-with-l2tpv3-Mjk0Nw%3D%3D\)](#) »

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 XR1 should provide connectivity between R1, R7, R8, and XR2.
 - Configure R2 and R4 with a spoke pseudowire to R3.
 - Configure R5 and XR1 with a spoke pseudowire to R6.
 - Configure a core pseudowire between R3 and R6 to provide service for the spoke pseudowires.
 - The configuration should only allow VLAN 100 between R1, R7, R8, and XR2 to be extended over the MPLS network.
- Once complete, R1, R7 and R8 should form OSPFv2 adjacencies and have IP reachability to each other's Loopback0 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  
!  
12 vfi VPLS_R1_R7_R8_XR2 manual  
vpn id 100  
bridge-domain 100  
neighbor 3.3.3.3 encapsulation mpls  
  
R3:  
12 vfi VPLS_R1_R7_R8_XR2 manual  
vpn id 100  
bridge-domain 100  
neighbor 4.4.4.4 encapsulation mpls no-split-horizon  
neighbor 2.2.2.2 encapsulation mpls no-split-horizon  
neighbor 6.6.6.6 encapsulation mpls  
  
R4:  
interface GigabitEthernet2  
no ip address  
speed 1000  
no negotiation auto  
service instance 100 ethernet  
encapsulation dot1q 100  
bridge-domain 100  
!  
12 vfi VPLS_R1_R7_R8_XR2 manual  
vpn id 100  
bridge-domain 100  
neighbor 3.3.3.3 encapsulation mpls  
  
R5:  
interface GigabitEthernet2  
no ip address  
speed 1000  
no negotiation auto  
service instance 100 ethernet  
encapsulation dot1q 100  
bridge-domain 100  
!  
12 vfi VPLS_R1_R7_R8_XR2 manual  
vpn id 100  
bridge-domain 100  
neighbor 6.6.6.6 encapsulation mpls
```

```
R6:
12 vfi VPLS_R1_R7_R8_XR2 manual
vpn id 100
bridge-domain 100
neighbor 5.5.5.5 encapsulation mpls no-split-horizon
neighbor 19.19.19.19 encapsulation mpls no-split-horizon
neighbor 3.3.3.3 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 6.6.6.6 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 3.3.3.3 encapsulation mpls
!
bridge-domain 100
member GigabitEthernet2 service-instance 100
```

```
member vfi VPLS_R1_R7_R8_XR2
```

```
R3:
```

```
l2vpn vfi context VPLS_R1_R7_R8_XR2
```

```
vpn id 100
```

```
member 6.6.6.6 encapsulation mpls
```

```
!
```

```
bridge-domain 100
```

```
member vfi VPLS_R1_R7_R8_XR2
```

```
member 4.4.4.4 100 encapsulation mpls
```

```
member 2.2.2.2 100 encapsulation mpls
```

```
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 3.3.3.3 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 6.6.6.6 encapsulation mpls
```

```
bridge-domain 100
```

```
member GigabitEthernet2 service-instance 100
```

```
member vfi VPLS_R1_R7_R8_XR2
```

```
R6:
```

```
l2vpn vfi context VPLS_R1_R7_R8_XR2
```

```
vpn id 100
```

```
member 3.3.3.3 encapsulation mpls
```

```
!
```

```
bridge-domain 100
```

```
member vfi VPLS_R1_R7_R8_XR2
```

```
member 19.19.19.19 100 encapsulation mpls
```

```
member 5.5.5.5 100 encapsulation mpls
```

Verification

Two scalability issues are exposed by a standard VPLS deployment.

- Control-plane signaling and configuration overhead of a full mesh of pseudowires between PEs servicing the same VPLS instance.
- Inefficient packet replication at the ingress PE, where multi-destination frames are flooded out of the full mesh of pseudowires.

A network with 50 PE routers all servicing the same VPLS instance would require 1225 pseudowires to be configured in order to be fully meshed. The formula is $n*(n-1)/2$, where n is the number of PEs in each VPLS instance. This is similar to the scalability issues experienced in an iBGP network with no Route-Reflectors or Confederations. Having to configure these many pseudowires causes significant configuration overhead. LDP signaling is also increased, as a full mesh of Targeted LDP sessions would need to be maintained. H-VPLS introduces a level of hierarchy by defining two types of PE routers; u-PE and n-PE, User facing PE and Network facing PE respectively. The core network consists of n-PE routers that have fully meshed pseudowires for each VPLS instance. The pseudowires between the n-PEs are referred to as `core pseudowires`, to which the standard split-horizon rules apply. In our example, R3 and R6 are the core routers acting as n-PEs, with a core pseudowire between them servicing the VPLS instance. If there were more n-PEs servicing the VPLS instance, a full mesh of core pseudowires would be needed between them.

The u-PEs at the edge of the network have the Access Circuit towards the customers, and connect to the n-PEs in the core via `spoke pseudowires`. R2, R4, R5, and XR1 are the u-PEs in this example. This hierarchical design negates the need for a full mesh of pseudowires between all u-PEs servicing the VPLS instance, as the u-PE can connect into the VPLS network via pseudowires to the upstream n-PEs. Note that ideally u-PEs would connect to multiple n-PEs for redundancy.

Recall that pseudowires have split-horizon enabled by default to prevent loops. In H-VPLS these rules are relaxed, by disabling split-horizon on the spoke pseudowires. If a frame coming in from a customer site is destined to a remote site that is reachable by transiting the core network, the frame is encapsulated by the u-PE onto the spoke pseudowire, which is received by an n-PE and switched out of the corresponding core pseudowire. Essentially the n-PE treats the spoke pseudowire as if it were an Access Circuit. From the u-PE's perspective, the pseudowire to the n-PE is a standard pseudowire with split-horizon enabled. It is the n-PE that disables split-horizon on it and treats it as a `spoke connection`. Going back to the iBGP analogy, the pseudowire connectivity model of H-VPLS can be compared to a Route-Reflector design where multiple RRs are fully meshed with iBGP sessions, and edge routers peer with each RR instead of employing a full mesh of sessions between all edge routers.

Notice that two sets of configurations were provided. Either one yields the desired network state, but both were given for completeness. The second set of configurations uses the newer

`MPLS L2VPN Protocol-Based CLI` referred to in the Cisco documentation for L2VPN (http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mp_l2_vpns/configuration/xr-3s/mp-l2-vpns-xr-3s-book/l2vpn-prot-based.html).

The final verification can be done from the customer's point of view, as we have been doing for all previous L2VPN tasks. The resulting state places R1, R7, and R8 in the same broadcast domain, being provided by the H-VPLS service. XR2 is not reachable over VPLS due to the XRv L2VPN data-plane limitation.

```
R1#show ip ospf neighbor
```

| Neighbor ID | Pri | State | Dead Time | Address | Interface |
|-------------|-----|--------------|-----------|------------|----------------------|
| 7.7.7.7 | 1 | FULL/DROTHER | 00:00:38 | 10.0.100.7 | GigabitEthernet2.100 |
| 8.8.8.8 | 1 | FULL/DR | 00:00:35 | 10.0.100.8 | GigabitEthernet2.100 |

```
R1#show ip route ospf
```

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP

a - application route

+ - replicated route, % - next hop override

Gateway of last resort is not set

7.0.0.0/32 is subnetted, 1 subnets

O 7.7.7.7 [110/2] via 10.0.100.7, 00:57:00, GigabitEthernet2.100

8.0.0.0/32 is subnetted, 1 subnets

O 8.8.8.8 [110/2] via 10.0.100.8, 00:57:00, GigabitEthernet2.100

```
R1#ping 7.7.7.7 source loo0
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 7.7.7.7, timeout is 2 seconds:

Packet sent with a source address of 1.1.1.1

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/9/23 ms

```
R1#ping 8.8.8.8 source loo0
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:

Packet sent with a source address of 1.1.1.1

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 2/9/22 ms

```
R1#show ip arp
```

| Protocol | Address | Age (min) | Hardware Addr | Type | Interface |
|----------|------------|-----------|----------------|------|----------------------|
| Internet | 10.0.100.1 | - | 0050.569e.35b6 | ARPA | GigabitEthernet2.100 |
| Internet | 10.0.100.7 | 80 | 0050.569e.0b8a | ARPA | GigabitEthernet2.100 |
| Internet | 10.0.100.8 | 76 | 0050.569e.30c3 | ARPA | GigabitEthernet2.100 |
| Internet | 10.0.200.1 | - | 0050.569e.35b6 | ARPA | GigabitEthernet2.200 |

Similar to the previous VPLS example, the pseudowires (both spoke and core) were signaled using LDP. Targeted LDP sessions are established, and the VC labels are exchanged accordingly. Notice that the VC ID is what is making all of these pseudowires part of the same VPLS instance.

R2#show mpls l2transport vc detail

Local interface: VFI VPLS_R1_R7_R8_XR2 vfi up

Interworking type is Ethernet

Destination address: 3.3.3.3, VC ID: 100, VC status: up

Output interface: Gi1.23, imposed label stack {28}

Preferred path: not configured

Default path: active

Next hop: 20.2.3.3

Create time: 1d22h, last status change time: 1d22h

Last label FSM state change time: 1d22h

Signaling protocol: LDP, peer 3.3.3.3:0 up

Targeted Hello: 2.2.2.2(LDP Id) -> 3.3.3.3, LDP is UP

Graceful restart: not configured and not enabled

Non stop routing: not configured and not enabled

Status TLV support (local/remote) : enabled/supported

LDP route watch : enabled

Label/status state machine : established, LruRru

Last local dataplane status rcvd: No fault

Last BFD dataplane status rcvd: Not sent

Last BFD peer monitor status rcvd: No fault

Last local AC circuit status rcvd: No fault

Last local AC circuit status sent: No fault

Last local PW i/f circ status rcvd: No fault

Last local LDP TLV status sent: No fault

Last remote LDP TLV status rcvd: No fault

Last remote LDP ADJ status rcvd: No fault

MPLS VC labels: local 29, remote 28

Group ID: local 0, remote 0

MTU: local 1500, remote 1500

Remote interface description:

Sequencing: receive disabled, send disabled

Control Word: On (configured: autosense)

SSO Descriptor: 3.3.3.3/100, local label: 29

Dataplane:

SSM segment/switch IDs: 8351/4252 (used), PWID: 1

VC statistics:

transit packet totals: receive 36123, send 18052

transit byte totals: receive 3686954, send 2310568

transit packet drops: receive 36076, seq error 0, send 0

R3#show mpls l2transport vc detail

Destination address: 2.2.2.2, VC ID: 100, VC status: up

Targeted Hello: 3.3.3.3(LDP Id) -> 2.2.2.2, LDP is UP

MPLS VC labels: local 28, remote 29

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

Targeted Hello: 3.3.3.3(LDP Id) -> 4.4.4.4, LDP is UP

MPLS VC labels: local 27, remote 26

Destination address: 6.6.6.6, VC ID: 100, VC status: up

Targeted Hello: 3.3.3.3(LDP Id) -> 6.6.6.6, LDP is UP

MPLS VC labels: local 29, remote 26

```
RP/0/0/CPU0:XR1#show l2vpn atom-db
Wed Jul  8 23:15:00.837 UTC
PW: neighbor 6.6.6.6, PW ID 100, state is up ( established )
PW class VPLS_CLASS, XC ID 0xff000001
```

```
Encapsulation MPLS, protocol LDP
Source address 19.19.19.19
PW type Ethernet, control word disabled, interworking none
PW backup disable delay 0 sec
Sequencing not set
```

PW Status TLV in use

| MPLS | Local | Remote |
|-----------------------|-------------------------|-------------------------|
| Label | 16013 | 29 |
| Group ID | unassigned | unknown |
| 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: 4278190081
Create time: 07/07/2015 00:28:25 (1d22h ago)
Last time status changed: 07/07/2015 00:28:26 (1d22h ago)
```

Configuration info:

```
PW class: VPLS_CLASS
Peer ID = 6.6.6.6, pseudowire ID = 100
Control word is not set
Transport mode: not set
Configured (Static) Encapsulation: not set
Provisioned Encapsulation: MPLS
```

<output snipped>

The VFI and bridge-domain constructs form the VPLS instance by housing the pseudowires and Access Circuits, and enabling data-plane based MAC learning on the PEs. Notice that for u-PE devices, the Bridge Domain includes the Access Circuit towards the CE, and the VFI only contains a single pseudowire towards the corresponding n-PE.

```
R2#show bridge-domain 100
Bridge-domain 100 (2 ports in all)
State: UP                Mac learning: Enabled
Aging-Timer: 300 second(s)
GigabitEthernet2 service instance 100
vfi VPLS_R1_R7_R8_XR2 neighbor 3.3.3.3 100
AED MAC address  Policy  Tag      Age  Pseudoport
0  0050.569E.35B6 forward dynamic  295  GigabitEthernet2.EFP100
0  0050.569E.0B8A forward dynamic  294  VPLS_R1_R7_R8_XR2.1001019
1  FFFF.FFFF.FFFF flood  static    0    OLIST_PTR:0xe84dc420
0  0050.569E.30C3 forward dynamic  291  VPLS_R1_R7_R8_XR2.1001019
```

```
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 | | | | | | |
| pw100008 | | VPLS_R1_R7_R8_XR2(VFI) | 0 | UP | UP | |
| pw100007 | core_pw | 3.3.3.3:100(MPLS) | 0 | UP | UP | |
| | | Local VC label 29 | | | | |
| | | Remote VC label 28 | | | | |
| BD name: 100, State: -- | | | | | | |
| - | | 100(BD) | 0 | UP | -- | |
| - | | VPLS_R1_R7_R8_XR2(VFI) | 0 | UP | UP | |

```
RP/0/0/CPU0:XR1#show l2vpn bridge-domain
Wed Jul  8 21:36:25.303 UTC
Legend: pp = Partially Programmed.
Bridge group: VPLS, bridge-domain: 100, id: 0, state: up, ShgId: 0, MSTi: 0
Aging: 300 s, MAC limit: 4000, Action: none, Notification: syslog
Filter MAC addresses: 0
ACs: 1 (1 up), VFIs: 1, PWs: 1 (1 up), PBBs: 0 (0 up)
List of ACs:
Gi0/0/0/1.100, state: up, Static MAC addresses: 0
List of Access PWs:
List of VFIs:
VFI 100 (up)
Neighbor 6.6.6.6 pw-id 100, state: up, Static MAC addresses: 0
```

As stated previously, the u-PE does not disable split-horizon on the pseudowire to the n-PE. This special treatment is only done by the n-PE.

```
R2#show 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:

Neighbors connected via pseudowires:

Peer Address      VC ID      S
3.3.3.3           100        Y
```

From an n-PEs point of view, the Bridge Domain contains spoke pseudowires, and the VFI contains the core pseudowires.

```
R3#show bridge-domain
Bridge-domain 100 (3 ports in all)

State: UP                Mac learning: Enabled
Aging-Timer: 300 second(s)

vfi VPLS_R1_R7_R8_XR2 neighbor 6.6.6.6 100
vfi VPLS_R1_R7_R8_XR2 neighbor 2.2.2.2 100
vfi VPLS_R1_R7_R8_XR2 neighbor 4.4.4.4 100

AED MAC address  Policy Tag      Age Pseudoport
0 0050.569E.35B6 forward dynamic 298 VPLS_R1_R7_R8_XR2.1001018
0 0050.569E.0B8A forward dynamic 295 VPLS_R1_R7_R8_XR2.1001019
1 FFFF.FFFF.FFFF flood static 0 OLIST_PTR:0xe8092c20
0 0050.569E.30C3 forward dynamic 299 VPLS_R1_R7_R8_XR2.1001017

R3#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
pw100011      VPLS_R1_R7_R8_XR2(VFI) 0 UP UP
pw100014      core_pw    4.4.4.4:100(MPLS) 0 UP UP
Local VC label 27
Remote VC label 26
pw100013      core_pw    2.2.2.2:100(MPLS) 0 UP UP
Local VC label 28
Remote VC label 29
pw100010      core_pw    6.6.6.6:100(MPLS) 0 UP UP
Local VC label 29
Remote VC label 26

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

The spoke pseudowires have split-horizon disabled on the n-PE, allowing frames to come in from a spoke pseudowire and being switched out of a core pseudowire.

```
R3#show vfi name VPLS_R1_R7_R8_XR2
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:
Neighbors connected via pseudowires:
Peer Address      VC ID      S
4.4.4.4           100        N
2.2.2.2           100        N
6.6.6.6           100        Y
```

The following table describes the split horizon rules of VPLS.

CONTENTS >

| From | To | Allowed |
|----------|----------------|---------|
| Spoke PW | Access Circuit | Yes |
| Spoke PW | Spoke PW | Yes |
| Spoke PW | Core PW | Yes |
| Core PW | Access Circuit | Yes |
| Core PW | Core PW | No |

Lets follow how R8 learns about R1's MAC address over the H-VPLS network:

R1 is using **0050.569e.35b6** on its Gig2.100 interface.

```
R1#show interfaces GigabitEthernet2.100 | inc bia
Hardware is CSR vNIC, address is 0050.569e.35b6 (bia 0050.569e.35b6 )
```

R2 learns this MAC over the EFP configured on the Access Circuit, Gig2.

```
R2#show bridge-domain 100
Bridge-domain 100 (2 ports in all)
State: UP           Mac learning: Enabled
Aging-Timer: 300 second(s)
GigabitEthernet2 service instance 100
vfi VPLS_R1_R7_R8_XR2 neighbor 3.3.3.3 100
AED MAC address  Policy Tag      Age Pseudoport
0  0050.569E.35B6 forward dynamic  296 GigabitEthernet2.EFP100
0  0050.569E.0B8A forward dynamic  297 VPLS_R1_R7_R8_XR2.1001019
1  FFFF.FFFF.FFFF flood static    0  OLIST_PTR:0xe84dc420
0  0050.569E.30C3 forward dynamic  300 VPLS_R1_R7_R8_XR2.1001019
```

R3 learns about **0050.569e.35b6** via its spoke pseudowire to R2.

<https://t.me/learningnets>

```
R3#show bridge-domain 100

Bridge-domain 100 (3 ports in all)

State: UP                Mac learning: Enabled

Aging-Timer: 300 second(s)

vfi VPLS_R1_R7_R8_XR2 neighbor 6.6.6.6 100

vfi VPLS_R1_R7_R8_XR2 neighbor 2.2.2.2 100

vfi VPLS_R1_R7_R8_XR2 neighbor 4.4.4.4 100

AED MAC address  Policy Tag      Age Pseudoport
0  0050.569E.35B6 forward dynamic  294 VPLS_R1_R7_R8_XR2.1001018
0  0050.569E.0B8A forward dynamic  292 VPLS_R1_R7_R8_XR2.1001019
1  FFFF.FFFF.FFFF flood  static  0  OLIST_PTR:0xe8092c20
0  0050.569E.30C3 forward dynamic  295 VPLS_R1_R7_R8_XR2.1001017
```

Since frames are allowed to pass between spoke and core pseudowires, R6 learns about **0050.569e.35b6** via its core pseudowire to R3.

```
R6#show bridge-domain 100

Bridge-domain 100 (3 ports in all)

State: UP                Mac learning: Enabled

Aging-Timer: 300 second(s)

vfi VPLS_R1_R7_R8_XR2 neighbor 3.3.3.3 100

vfi VPLS_R1_R7_R8_XR2 neighbor 5.5.5.5 100

vfi VPLS_R1_R7_R8_XR2 neighbor 19.19.19.19 100

AED MAC address  Policy Tag      Age Pseudoport
0  0050.569E.35B6 forward dynamic  296 VPLS_R1_R7_R8_XR2.1001013
0  0050.569E.0B8A forward dynamic  295 VPLS_R1_R7_R8_XR2.1001013
1  FFFF.FFFF.FFFF flood  static  0  OLIST_PTR:0xe80d7420
0  0050.569E.30C3 forward dynamic  298 VPLS_R1_R7_R8_XR2.1001014
```

Following the sequence, u-PE router R5 learns about **0050.569e.35b6** via its pseudowire to R6.

```
R5#show bridge-domain 100

Bridge-domain 100 (2 ports in all)

State: UP                Mac learning: Enabled

Aging-Timer: 300 second(s)

GigabitEthernet2 service instance 100

vfi VPLS_R1_R7_R8_XR2 neighbor 6.6.6.6 100

AED MAC address  Policy Tag      Age Pseudoport
0  0050.569E.35B6 forward dynamic  290 VPLS_R1_R7_R8_XR2.1001025
0  0050.569E.0B8A forward dynamic  292 VPLS_R1_R7_R8_XR2.1001025
1  FFFF.FFFF.FFFF flood  static  0  OLIST_PTR:0xe80c8860
0  0050.569E.30C3 forward dynamic  292 GigabitEthernet2.EFP100
```

Finally, R8 sees **0050.569e.35b6** in its ARP cache.

```
R8#show ip arp

Protocol Address          Age (min) Hardware Addr  Type  Interface
Internet 10.0.100.1         55  0050.569e.35b6 ARPA  GigabitEthernet2.100
Internet 10.0.100.7         39  0050.569e.0b8a ARPA  GigabitEthernet2.100
Internet 10.0.100.8         -   0050.569e.30c3 ARPA  GigabitEthernet2.100
Internet 10.0.200.8         -   0050.569e.30c3 ARPA  GigabitEthernet2.200
```

CONTENTS ▼