MPLS L2 VPN - Port Based Point-to-Point Service

« MPLS L3VPN 6VPE (/workbook/view/service-provider-v4/task/mpls-l3vpn-6vpe-Mjg3Mw%3D%3D) | MPLS L2 VPN - Port Based Point-to-Point Service using EVC (/workbook/view/service-provider-v4/task/mpls-l2-vpn-port-based-point-to-point-service-using-evc-Mjk0Mw%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 L2VPN with Any Transport over MPLS as follows:
  - The attachment circuits are the Ethernet links connecting to R1, R7, R8 and XR2 respectively.
  - R2 and XR1 should form a pseudowire between their Loopback0 interfaces.
  - R4 and R5 should form a pseudowire between their Loopback0 interfaces.
- Once complete, R7 and R8 should form an OSPFv2 adjacency and have IP reachability to each other's Looopback0 networks.
- Note that R1 and XR1 will not form an OSPFv2 adjacency due to the XRv L2VPN limitation.

Configuration **Click to *collapse***

```
R2:
pseudowire-class ETH_TO_ETH
 encapsulation mpls
!
interface GigabitEthernet2
 xconnect 19.19.19.19 219 pw-class ETH_TO_ETH


R4:
pseudowire-class ETH_TO_ETH
 encapsulation mpls
!
interface GigabitEthernet2
 xconnect 5.5.5.5 45 pw-class ETH_TO_ETH



R5:
pseudowire-class ETH_TO_ETH
 encapsulation mpls
!
interface GigabitEthernet2
 xconnect 4.4.4.4 45 pw-class ETH_TO_ETH



XR1:
interface GigabitEthernet0/0/0/1
 no cdp
 l2transport
 !
!
l2vpn
 pw-class ETH_TO_ETH
  encapsulation mpls
  !
 !
 xconnect group GROUP1
  p2p XR1_R2
   interface GigabitEthernet0/0/0/1
   neighbor ipv4 2.2.2.2 pw-id 219
    pw-class ETH_TO_ETH
   !
  !
 !
!
```

Verification

MPLS L2VPN differs from L3VPN in the fact that the customer does not peer Layer 3 Routing with the Service Provider. Instead, customer sites are on the same emulated layer 2 network, similar to legacy Frame Relay or ATM networks. This means that the customer's Layer 3 Routing happens as an overlay on top of the L2VPN, and all details of the Service Provider MPLS network are completely transparent to the customer. This technology is used by the typical Service Provider Point-to-Point Metro-Ethernet offerings. To the customer it looks as if their routers/switches at each end of the circuit are directly connected.

The first verification for MPLS L2VPN with AToM is to ensure that an LSP can be formed between the PE routers' Loopback interfaces. This is similar to the requirement of L3VPN forming a VPNv4 BGP peering over an LSP between PE Loopbacks, but with AToM the labels for the L2VPN circuit are allocated via targeted LDP sessions. Note that the underlying network still needs to be running some sort of labeling, such as LDP or MPLS-TE.

The traceroute below verifies that an LSP does exist between the PE's Loopback interfaces. This is the LSP established by the standard LDP sessions running throughout the core. We could have used MPLS-TE for this as well.

```
R2#traceroute 19.19.19.19 source lo0


Type escape sequence to abort.
Tracing the route to 19.19.19.19
VRF info: (vrf in name/id, vrf out name/id)
  1 20.2.3.3 [MPLS: Label 26 Exp 0] 68 msec
    20.2.4.4 [MPLS: Label 26 Exp 0] 8 msec
    20.2.3.3 [MPLS: Label 26 Exp 0] 6 msec
  2 20.4.5.5 [MPLS: Label 27 Exp 0] 2 msec
    20.3.6.6 [MPLS: Label 25 Exp 0] 3 msec
    20.4.5.5 [MPLS: Label 27 Exp 0] 1 msec
  3 20.6.19.19 14 msec
    20.5.19.19 16 msec *
```

R2 is receiving label 26 from R3 and R4 to reach 19.19.19.19/32.

```
R2#show ip cef 19.19.19.19/32 detail
19.19.19.19/32, epoch 2, per-destination sharing
  local label info: global/28
  1 RR source [no flags]
   nexthop 20.2.3.3 GigabitEthernet1.23 label 26
   nexthop 20.2.4.4 GigabitEthernet1.24 label 26


R2#show mpls ldp bindings remote-label 26
  lib entry: 19.19.19.19/32, rev 30
        remote binding: lsr: 4.4.4.4:0, label: 26
        remote binding: lsr: 3.3.3.3:0, label: 26
```

The next step in AToM is to establish the Pseudowire adjacency between the PE routers. This occurs automatically once the **xconnect** is configured on the Attachment Circuit, which is the CE facing link. If the **xconnect** is successful the PE routers should form an LDP adjacency, as seen below.

```
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#pseudowire-class ETH_TO_ETH
R2(config-pw-class)# encapsulation mpls
R2(config-pw-class)#!
R2(config-pw-class)#interface GigabitEthernet2
R2(config-if)# xconnect 19.19.19.19 216 pw-class ETH_TO_ETH
R2(config-if-xconn)#end
%SYS-5-CONFIG_I: Configured from console by console
R2#
 %LDP-5-NBRCHG: LDP Neighbor 19.19.19.19:0 (3) is UP



R2#show mpls ldp neighbor 19.19.19.19 detail
    Peer LDP Ident: 19.19.19.19:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 19.19.19.19.21095 - 2.2.2.2.646
        Password: not required, none, in use
        State: Oper; Msgs sent/rcvd: 35/35; Downstream; Last TIB rev sent 30
        Up time: 00:10:23; UID: 4; Peer Id 2
        LDP discovery sources:
          Targeted Hello 2.2.2.2 -> 19.19.19.19, active, passive;
            holdtime: infinite, hello interval: 10000 ms
        Addresses bound to peer LDP Ident:
          19.19.19.19     20.6.19.19      20.5.19.19
        Peer holdtime: 180000 ms; KA interval: 60000 ms; Peer state: estab
        Clients: Dir Adj Client
        NSR: Not Ready
        Capabilities Sent:
          [ICCP (type 0x0405) MajVer 1 MinVer 0]
          [Dynamic Announcement (0x0506)]
          [mLDP Point-to-Multipoint (0x0508)]
          [mLDP Multipoint-to-Multipoint (0x0509)]
          [Typed Wildcard (0x050B)]
        Capabilities Received:
          [mLDP Point-to-Multipoint (0x0508)]
          [mLDP Multipoint-to-Multipoint (0x0509)]
          [Typed Wildcard (0x050B)]


RP/0/0/CPU0:XR1#show mpls ldp neighbor 2.2.2.2 detail
Thu Jun 11 23:03:32.610 UTC


Peer LDP Identifier: 2.2.2.2:0
  TCP connection: 2.2.2.2:646 - 19.19.19.19:21095
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 36/36; Downstream-Unsolicited
  Up time: 00:11:16
  LDP Discovery Sources:
      Targeted Hello (19.19.19.19 -> 2.2.2.2, active)
  Addresses bound to this peer:
      2.2.2.2         20.2.3.2        20.2.4.2
  Peer holdtime: 180 sec; KA interval: 60 sec; Peer state: Estab
```

```
NSR: Disabled
Clients: AToM
Capabilities:
  Sent:
    0x508  (MP: Point-to-Multipoint (P2MP))
    0x509  (MP: Multipoint-to-Multipoint (MP2MP))
    0x50b  (Typed Wildcard FEC)
  Received:
    0x508  (MP: Point-to-Multipoint (P2MP))
    0x509  (MP: Multipoint-to-Multipoint (MP2MP))
    0x50b  (Typed Wildcard FEC)
```

This targeted LDP session is then used between the PE routers to exchange the Pseudowire Label, also referred to as the VC label (Virtual Circuit), which is analogous to the VPNv4 BGP Label in MPLS L3VPN. The Pseudowire Label is what the PE routers use to determine which Attachment Circuit traffic should forwarded towards when labeled packets arrive in the data plane from the Service Provider network. This could likewise be considered the “L2VPN Label”, and has a similar "multiplexing" function as a VPNv4 label in L3VPN. This label will be used at the bottom of the stack, while the transport label between the Pseudowire endpoints (the PE routers’ Loopbacks) will be on the top of the stack. This label number can be verified just like a normal L3VPN label as follows.

```
R2#show mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         Pop Label  3.3.3.3/32       58            Gi1.23     20.2.3.3
17         Pop Label  20.3.4.0/24      0             Gi1.23     20.2.3.3
           Pop Label  20.3.4.0/24      0             Gi1.24     20.2.4.4
18         Pop Label  20.3.6.0/24      0             Gi1.23     20.2.3.3
19         Pop Label  4.4.4.4/32       676           Gi1.24     20.2.4.4
20         Pop Label  20.4.6.0/24      0             Gi1.24     20.2.4.4
21         Pop Label  20.4.5.0/24      0             Gi1.24     20.2.4.4
22         16         5.5.5.5/32       0             Gi1.24     20.2.4.4
23         19         20.5.19.0/24     0             Gi1.24     20.2.4.4
24         23         20.5.6.0/24      0             Gi1.23     20.2.3.3
           20         20.5.6.0/24      0             Gi1.24     20.2.4.4
25         24         6.6.6.6/32       0             Gi1.23     20.2.3.3
           23         6.6.6.6/32       0             Gi1.24     20.2.4.4
26         25         20.6.19.0/24     0             Gi1.23     20.2.3.3
           24         20.6.19.0/24     0             Gi1.24     20.2.4.4
28         26         19.19.19.19/32   0             Gi1.23     20.2.3.3
           26         19.19.19.19/32   0             Gi1.24     20.2.4.4
29         No Label   l2ckt(3)         0             Gi2        point2point
```

R2 has allocated local label 29 for virtual circuit ID 219, and has received label 16000 from XR1.

```
R2#show xconnect peer 19.19.19.19 vcid 219 detail

Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State

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

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


XC ST  Segment 1                         S1 Segment 2                         S2

------+--------------------------------+--+--------------------------------+--

UP pri    ac Gi2:7(Ethernet)              UP mpls  19.19.19.19:219              UP

            Interworking: none                    Local VC label 29

                                                  Remote VC label 16000

                                                 pw-class: ETH_TO_ETH


RP/0/0/CPU0:XR1#show mpls forwarding

Thu Jun 11 23:08:24.010 UTC

Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes

Label  Label       or ID              Interface                    Switched

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

 16000  Pop         PW(2.2.2.2:219)    Gi0/0/0/1    point2point     0

16001  18          2.2.2.2/32         Gi0/0/0/0.519 20.5.19.5       10100

       19          2.2.2.2/32         Gi0/0/0/0.619 20.6.19.6       444

16002  18          3.3.3.3/32         Gi0/0/0/0.619 20.6.19.6       0

16003  16          4.4.4.4/32         Gi0/0/0/0.519 20.5.19.5       0

       17          4.4.4.4/32         Gi0/0/0/0.619 20.6.19.6       0

16004  Pop         5.5.5.5/32         Gi0/0/0/0.519 20.5.19.5       1764

16005  Pop         6.6.6.6/32         Gi0/0/0/0.619 20.6.19.6       1822

16006  Pop         20.5.6.0/24        Gi0/0/0/0.519 20.5.19.5       0

       Pop         20.5.6.0/24        Gi0/0/0/0.619 20.6.19.6       0

16007  Pop         20.4.6.0/24        Gi0/0/0/0.619 20.6.19.6       0

16008  Pop         20.3.6.0/24        Gi0/0/0/0.619 20.6.19.6       0

16009  24          20.2.3.0/24        Gi0/0/0/0.619 20.6.19.6       0

16010  Pop         20.4.5.0/24        Gi0/0/0/0.519 20.5.19.5       0

16011  22          20.2.4.0/24        Gi0/0/0/0.519 20.5.19.5       0

       22          20.2.4.0/24        Gi0/0/0/0.619 20.6.19.6       0

16012  23          20.3.4.0/24        Gi0/0/0/0.519 20.5.19.5       0

       21          20.3.4.0/24        Gi0/0/0/0.619 20.6.19.6       0
```

Likewise, XR1 has allocated local label 16000 for virtual circuit ID 219, and has received label 29 from R2. This exchange took place via the Targeted-LDP session.

```
RP/0/0/CPU0:XR1#show l2vpn xconnect neighbor 2.2.2.2 detail
Thu Jun 11 23:17:41.832 UTC

Group GROUP1, XC XR1_R2, state is up; Interworking none
  AC: GigabitEthernet0/0/0/1, state is up
    Type Ethernet
    MTU 1500; XC ID 0x1; interworking none
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
  PW: neighbor 2.2.2.2, PW ID 219, state is up ( established )
    PW class ETH_TO_ETH, 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        16000                          29
      Group ID     0x200                          unknown
      Interface    GigabitEthernet0/0/0/1         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
    Outgoing Status (PW Status TLV):
      Status code: 0x0 (Up) in Notification message
    MIB cpwVcIndex: 4278190081
    Create time: 11/06/2015 22:49:49 (00:27:52 ago)
    Last time status changed: 11/06/2015 22:52:22 (00:25:19 ago)
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
```

Note that although the label exchange takes place between R2 and XR1 (the control-plane), the forwarding tables on XR1 are not programmed due to the implementation limitation noted at the beginning of the section, and thus packets will not be forwarded accross the Pseudowire on XR1. We will perform the data-plane verification on the Pseudowire between R4 and R5.

```
RP/0/0/CPU0:XR1#show l2vpn forwarding neighbor 2.2.2.2 pw-id 219 detail location 0/0/CPU0
Thu Jun 11 23:24:09.076 UTC
Local interface: GigabitEthernet0/0/0/1, Xconnect id: 0x1,  Status: down
  Segment 1
    AC, GigabitEthernet0/0/0/1, Ethernet port mode, status: Not bound
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
  Segment 2
    MPLS, Destination address: 2.2.2.2, pw-id: 219, status: Not bound
    Pseudowire label: 29    Control word disabled
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
```

To continue with the data-plane verification, shutdown the link between R4 and R5 so that we are able to see both transport and L2VPN labels in the stack through the core. Otherwise, forwarding between R4 and R5 will only include the VC label, as the transport label will be Implicit-Null due to the PEs being directly connected.

```
R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#interface Gig1.45
R4(config-subif)#shut
R4(config-subif)#end
R4#
%OSPF-5-ADJCHG: Process 1, Nbr 5.5.5.5 on GigabitEthernet1.45 from FULL to DOWN, Neighbor Down: Interface down or detached
%SYS-5-CONFIG_I: Configured from console by console
```

Note that the LSP between R4 and R5 now traverses R6.

```
R4#traceroute 5.5.5.5 source loopback 0
Type escape sequence to abort.
Tracing the route to 5.5.5.5
VRF info: (vrf in name/id, vrf out name/id)
  1 20.4.6.6 [MPLS: Label 16 Exp 0] 4 msec 2 msec 1 msec
  2 20.5.6.5 1 msec *  2 msec
```

R4 assigned local label 25 for the virtual circuit ID 45, and received label 26 from R5 via the targeted LDP session. The Pseudowire between R4 and R5 is active and the forwarding tables have been programmed.

```
R4#show xconnect peer 5.5.5.5 vcid 45 detail
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                         S2
------+---------------------------------+--+---------------------------------+--
UP pri    ac Gi2:7(Ethernet)               UP  mpls 5.5.5.5:45                   UP
            Interworking: none                    Local VC label 25
                                                  Remote VC label 26

                                            pw-class: ETH_TO_ETH
R4#show mpls l2transport pwid
AToM Pseudowire IDs: In use: 1, In holddown: 0

Label  Peer-Address    VCID       PWID       In-Use FirstUse ReusedAt FreedAt
------ --------------- ---------- ---------- ------ -------- -------- --------
 25     5.5.5.5         45         1          Yes    00:10:05 Never    Never




R5#show xconnect peer 4.4.4.4 vcid 45 detail
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                         S2
------+---------------------------------+--+---------------------------------+--
UP pri    ac Gi2:7(Ethernet)               UP  mpls 4.4.4.4:45                   UP
            Interworking: none                    Local VC label 26
                                                  Remote VC label 25
                                                 pw-class: ETH_TO_ETH

R5#show mpls l2transport pwid
AToM Pseudowire IDs: In use: 1, In holddown: 0

Label  Peer-Address    VCID       PWID       In-Use FirstUse ReusedAt FreedAt
------ --------------- ---------- ---------- ------ -------- -------- --------
 26     4.4.4.4         45         1          Yes    00:16:39 Never    Never
```

R7 and R8 should have established an OSPFv2 session, as if they were connected on the same wire/LAN. Their traffic is being tunneled over the L2VPN service being provided by R4 and R5.

```
R7#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
8.8.8.8           1   FULL/DR         00:00:34    10.0.0.8        GigabitEthernet2

R7#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

      8.0.0.0/32 is subnetted, 1 subnets
O        8.8.8.8 [110/2] via 10.0.0.8, 00:00:37, GigabitEthernet2



R7#traceroute 8.8.8.8 source 7.7.7.7
Type escape sequence to abort.
Tracing the route to 8.8.8.8
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.0.8 5 msec *  2 msec
```

Lets get a glimpse of what happens in the data-plane as the packets are encapsulated onto the L2VPN by enabling the FIA Trace feature on R4, and sending some packets to R8 from R7.

Note:

Note: The FIA (Forwarding Invocation Array) feature is documented here. (http://www.cisco.com/c/en/us/support/docs/content-networking/adaptive-session-redundancy-asr/117858-technote-asr-00.html)

```
R4#debug platform packet-trace enable
R4#debug platform packet-trace packet 128 fia-trace data-size 2048
R4#debug platform condition interface gig2 ingress
R4#debug platform condition start



R7#ping 8.8.8.8 source 7.7.7.7
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 7.7.7.7
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/20 ms
```

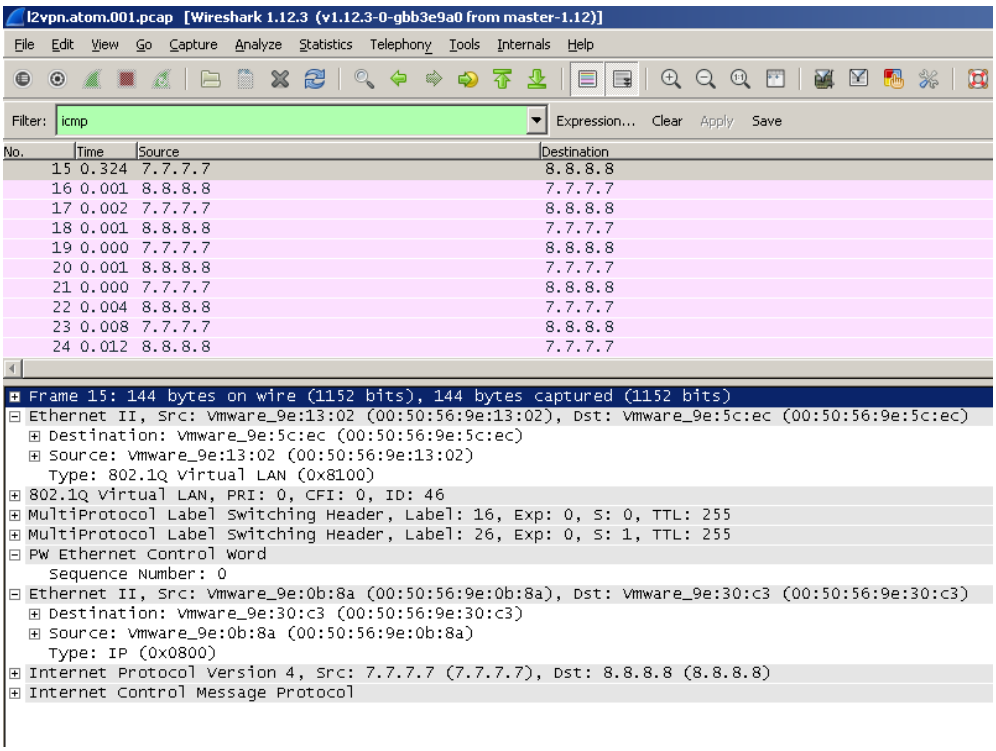
The packet comes into R4's interface Gig2, the Access Circuit, and invokes the L2VPN ATOM forwarding pipeline. Notice that the encapsulation type of the payload is gleaned from the packet, in this case ARPA.

```
R4#show platform packet-trace packet 10

Packet: 10          CBUG ID: 25

Summary

   Input     : GigabitEthernet2

   Output    : GigabitEthernet1

  State     : FWD

  Timestamp

    Start   : 5293293694976 ns (06/12/2015 00:09:24.098099 UTC)

    Stop    : 5293293722379 ns (06/12/2015 00:09:24.098126 UTC)

Path Trace

   Feature: IPV4

     Source      : 7.7.7.7

     Destination : 8.8.8.8

     Protocol    : 1 (ICMP)

  Feature: FIA_TRACE

    Entry      : 0x80880270 - DEBUG_COND_INPUT_PKT

    Lapsed time: 4240 ns

  Feature: FIA_TRACE

    Entry      : 0x80188220 - LAYER2_IPV4_INPUT_ARL_SANITY

    Lapsed time: 9333 ns

  Feature: FIA_TRACE

    Entry      : 0x80419db0 - LAYER2_IPV4_INPUT_MARTIAN

    Lapsed time: 1946 ns

   Feature: ATOM

    Feature name :  LAYER2_INPUT_XCONNECT PERF

    ifname       :  GigabitEthernet2

    xconnect id  : 0x7

    encap type   :  ARPA

    iw type      : LIKE2LIKE

     xconnect type: MPLS_PSEUDO_WIRE

  Feature: FIA_TRACE

    Entry      : 0x8076d720 - LAYER2_INPUT_XCONNECT

    Lapsed time: 41306 ns

  Feature: FIA_TRACE

    Entry      : 0x800af6b0 - MPLS_INPUT_GOTO_OUTPUT_FEATURE

    Lapsed time: 8506 ns

  Feature: FIA_TRACE

    Entry      : 0x8045bd00 - IPV4_VFR_REFRAG

    Lapsed time: 2880 ns

  Feature: FIA_TRACE

    Entry      : 0x80466d20 - IPV6_MC_INPUT_VFR_REFRAG

    Lapsed time: 1386 ns

  Feature: FIA_TRACE

    Entry      : 0x807bf9a0 -  MPLS_OUTPUT_ADD_LABEL

    Lapsed time: 3173 ns

  Feature: FIA_TRACE

    Entry      : 0x807b7030 - IPV6_OUTPUT_L2_REWRITE

    Lapsed time: 8960 ns

  Feature: FIA_TRACE

    Entry      : 0x804aadf0 - MPLS_OUTPUT_FRAG

    Lapsed time: 4320 ns

  Feature: FIA_TRACE

    Entry      : 0x8060bc80 - MPLS_OUTPUT_DROP_POLICY
```

```
    Lapsed time: 15466 ns

  Feature: FIA_TRACE

    Entry      : 0x80954900 - MARMOT_SPA_D_TRANSMIT_PKT

    Lapsed time: 35893 ns
```

Lets do a packet capture on R6 to see the entire label stack as it traverses the core network:

```
R6#monitor capture CAP match any interface Gig1.46 both

R6#monitor capture CAP start

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



R7#ping 8.8.8.8 source 7.7.7.7

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 7.7.7.7

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/20 ms



R6#monitor capture CAP stop

R6#

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


R6#monitor capture CAP export ftp://cisco:cisco@169.254.254.1/l2vpn.atom.001.pcap

Writing l2vpn.atom.pcap

Exported Successfully
```



In the packet capture above we see the ICMP echos between R7 and R8. Notice that a router in the core can easily look into these packets as they are not encrypted. The packet is formatted as follows:

- Outer Ethernet Header - Source and Dest MACs of the Gig1.46 link between R4 and R6
- Outer 802.1Q VLAN Tag - VLAN 46, used on the Gig1.46 link
- Transport Label - Label 16, advertised by R6 towards R4 to reach 5.5.5.5/32

- L2VPN Label - Label 26, advertised by R5 to R4 for VC-ID 45 via the Targeted LDP session
- PseudoWire CW - Control Word, currently not set.
- Inner Ethernet Header - Source R7's Gig2 MAC, Destination R8's Gig2 MAC.
- IP Header - Source 7.7.7.7, Destination 8.8.8.8
- ICMP Header - Echo Request

Notice that the entire frame is encapsulated and forwarded into the Pseudowire as it comes into R4's Access Circuit . This means that the Layer-2 frame is decapsulated by R5 and forwarded onto the AC. R8 should have an ARP entry for R7.


```
R8#show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.0.0.7              139   0050.569e.0b8a  ARPA   GigabitEthernet2
Internet  10.0.0.8                -   0050.569e.30c3  ARPA   GigabitEthernet2
```

The Control Word seen in the packet is not enabled by default (set to 0 in the packet), and it is not required for establishing a Pseudowire when the emulated circuit is Ethernet. Note that it is mandatory for other encapsulations, such as Frame Relay and ATM, as it carries Layer-2 specific control-bits, such as the FECN and BECN in Frame-Relay. By carrying this extra Layer-2 info, MPLS is able to emulate circuits for these encapsulation types. Enabling the Control-Word is recommended for Ethernet Pseudowires however, as it plays an important role in ECMP Load Sharing in the core of the network.

Normally PE routers look into the packet headers to determine whether a packet is IPv4 or IPv6, and then look at the source/destination tuples to make a load sharing decision. In the case of a Layer-2 VPN, the header that a PE router will inspect in Ethernet header instead of an IP header. If the MAC address happens to start with a 0x4 or a 0x6, the PE router will think it is dealing with an IPv4 or IPv6 packet and will mistakenly try to make a load sharing decision based on the wrong fields. By adding the control word, when the PE router tries to determine whether it is IPv4 or IPv6, it will look at the control word and find a non 0x4 or 0x6 field, thus it is guaranteed that it will not try to incorrectly load share the traffic. Notice where the control-word is placed in the packet. Its in the same location as a normal IP header would be in a L3VPN or standard L3 Forwarding. If the control-word is missing, the MAC is "exposed" to the router when making the look-up.

« MPLS L3VPN 6VPE (/workbook/view/service-provider-v4/task/mpls-l3vpn-6vpe-Mjg3Mw%3D%3D) | MPLS L2 VPN - Port Based Point-to-Point Service using EVC (/workbook/view/service-provider-v4/task/mpls-l2-vpn-port-based-point-to-point-service-using-evc-Mjk0Mw%3D%3D) »