Deploying MPLS L2VPN

Nurul Islam Roman (nurul@apnic.net)

BUILT FOR THE HUMAN NETWORK

CISCO

Abstract

- This session covers the fundamental and advanced topics associated with the deployment of Layer 2 VPNs over an MPLS network.
- The material presents a technology overview with an emphasis on ethernet-based point-to-point and multipoint VPNs. Session content then focuses on deployment considerations including: Signaling/Auto-discovery, OAM, Resiliency and Inter-AS.
- The attendee can expect to see sample configurations (IOS and IOS-XR) associated with the provisioning of L2VPNs.
- This session is intended for service providers and enterprise customers deploying L2VPNs over their MPLS network.

Agenda

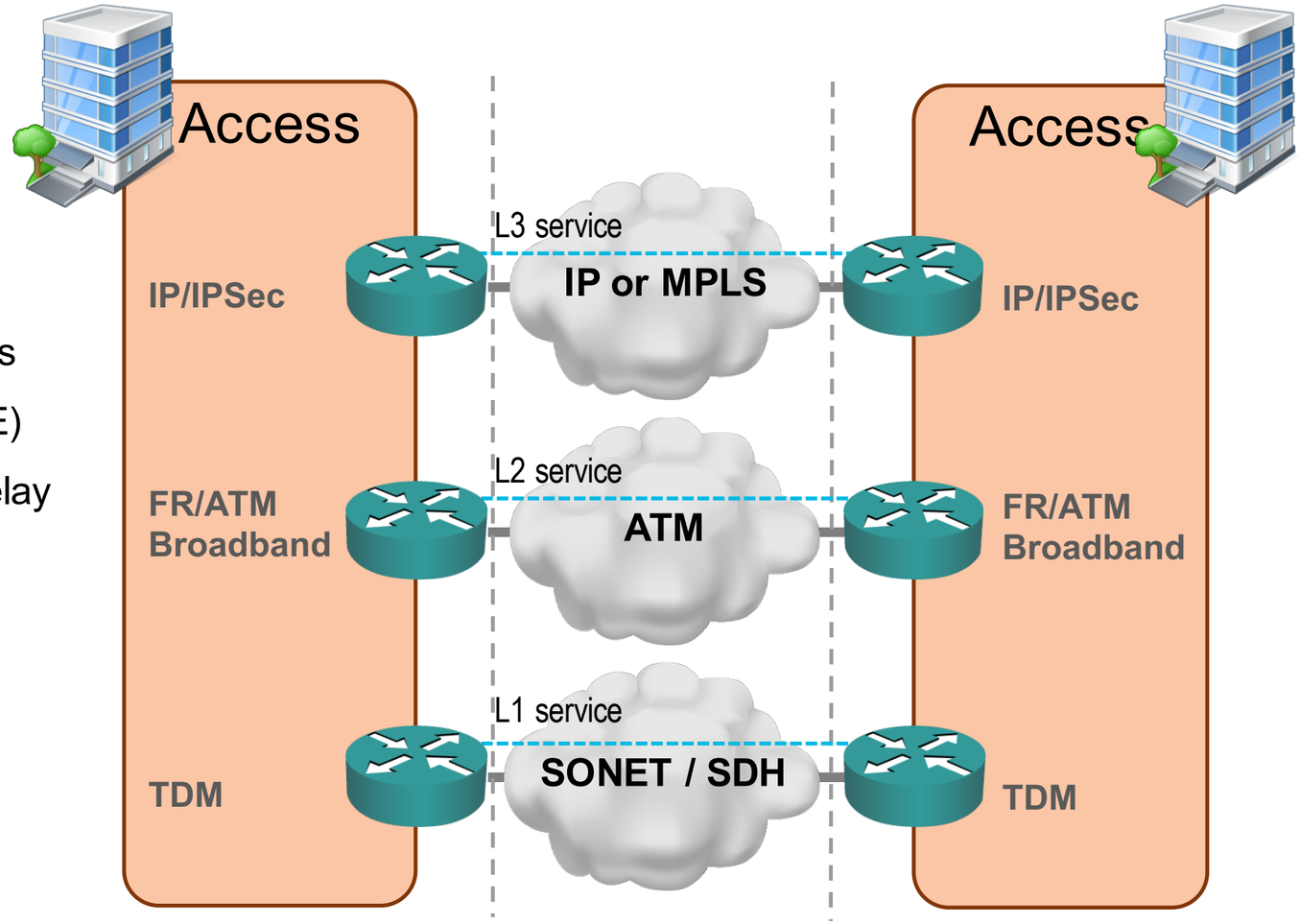
- Layer 2 VPN Motivation and Overview
- VPWS Reference Model
- VPLS Reference Model
- Pseudowire (PW) Signaling and PE Auto-Discovery
- Advanced Topics
- Summary

L2VPN Motivation and Overview

Motivation for L2VPNs

Old and New Drivers

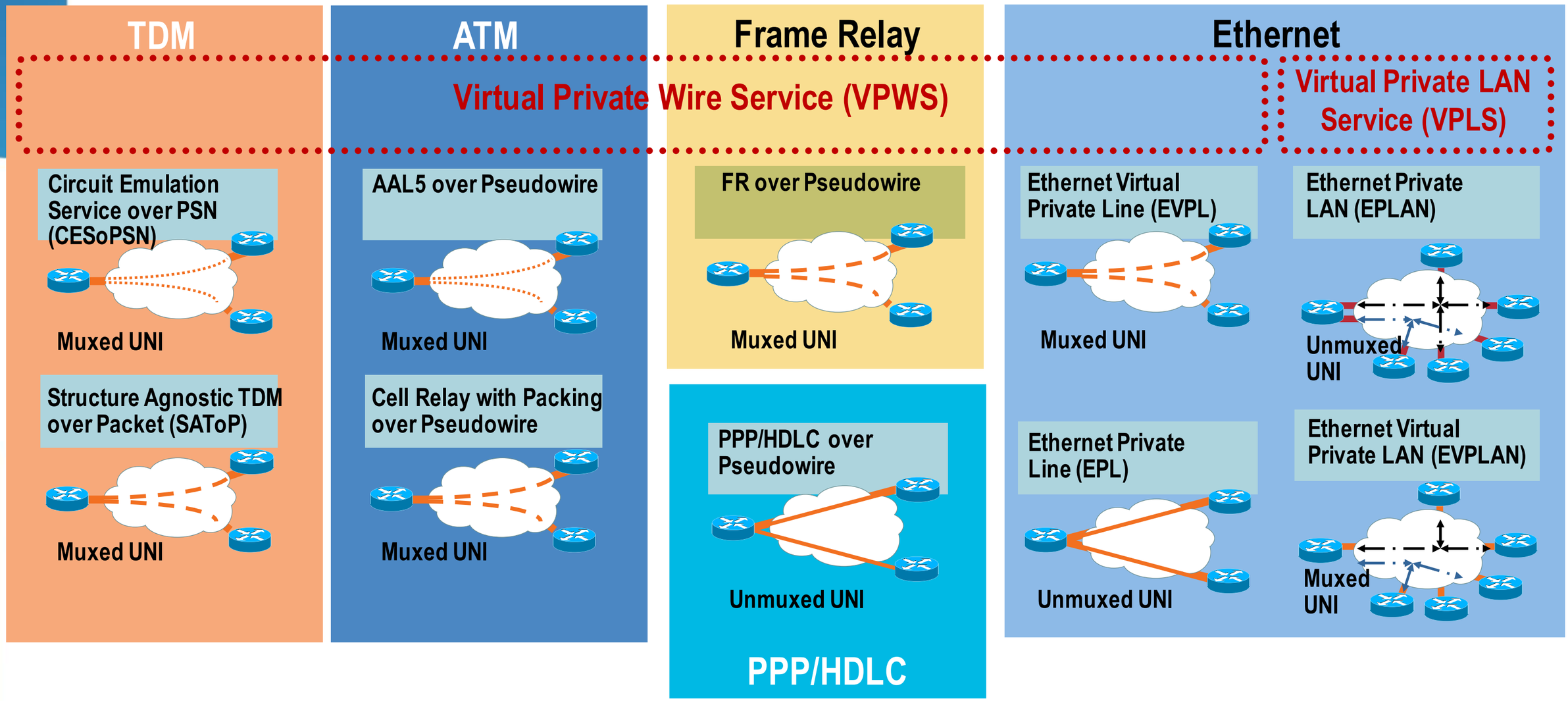
- Network Consolidation
  - Multiple access services (FR, ATM, TDM) required multiple core technologies
- Enterprise Ethernet WAN Connectivity Services
  - Ethernet well understood by Enterprise / SPs
  - CAPEX (lower cost per bit) / Growth (100GE)
  - Layer 2 VPN replacement to ATM/Frame Relay
  - Internet / Layer 3 VPN access (CE to PE)
- Data Center Interconnection (DCI)
- Mobile Backhaul Evolution
  - TDM /PDH to Dual/Hybrid to All-packet (IP/Ethernet)
  - Single (voice + data) IP/Ethernet mobile backhaul universally accepted solution



Typical Service Provider (circa 2000)

Service Offerings

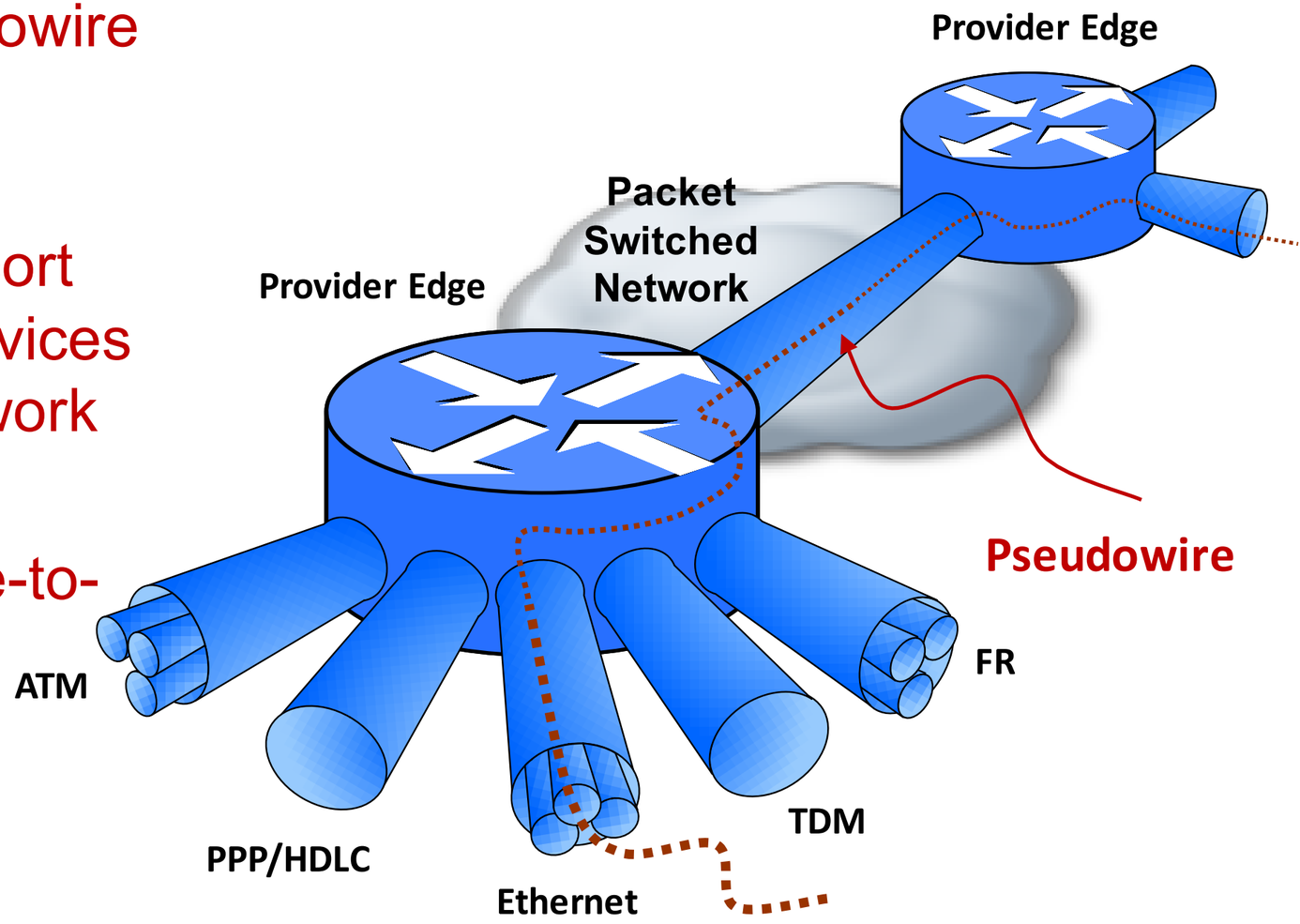
L2VPN Transport Services



Layer 2 VPN Enabler

The Pseudowire

- L2VPNs are built with Pseudowire (PW) technology
- PWs provide a common intermediate format to transport multiple types of network services over a Packet Switched Network (PSN)
- PW technology provides Like-to-Like transport and also Interworking (IW)

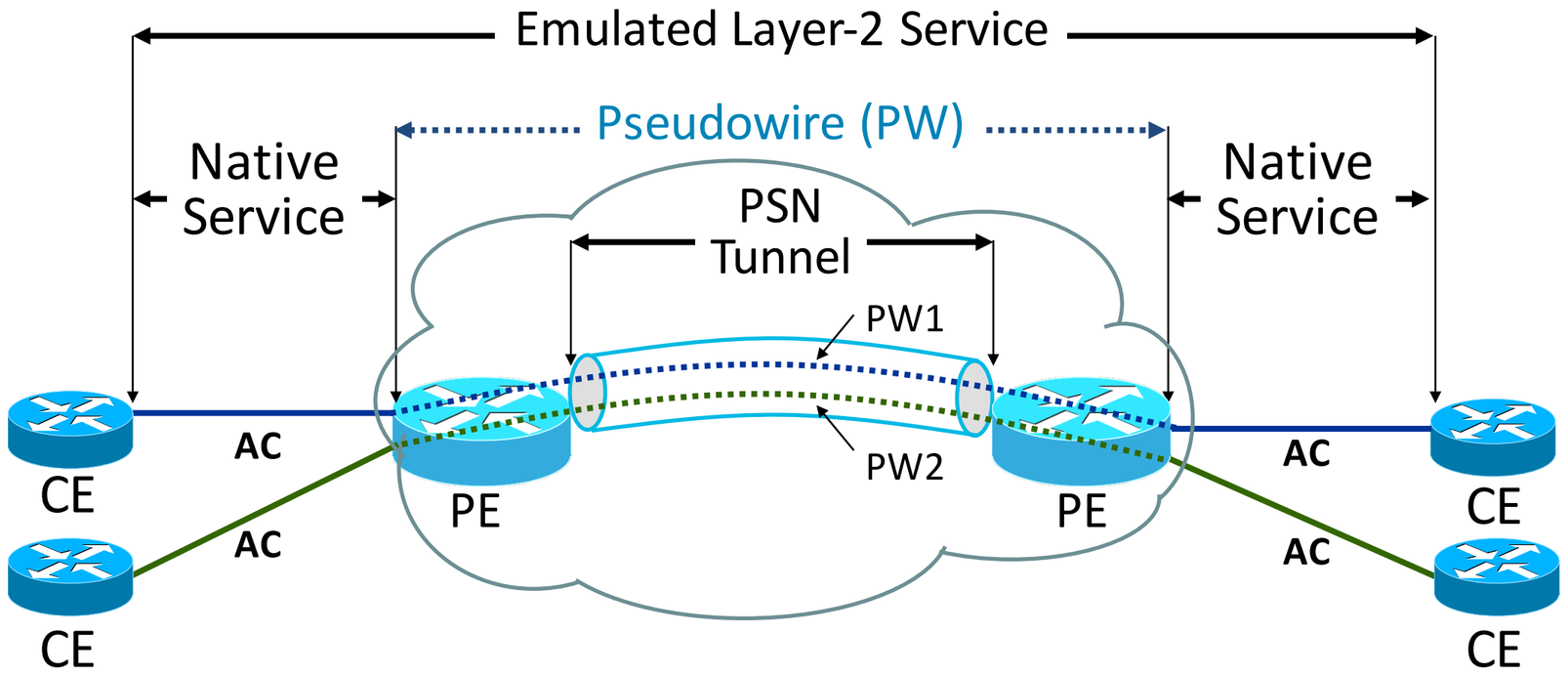


Virtual Private Wire Service (VPWS)

Overview

Pseudowire Reference Model

- Any Transport Over MPLS (AToM) is Cisco's implementation of VPWS for IP/MPLS networks
- An Attachment Circuit (AC) is the physical or virtual circuit attaching a CE to a PE
- Customer Edge (CE) equipment perceives a PW as an unshared link or circuit



Ref: RFC 3985 Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture, March 2005

Layer 2 Transport over MPLS

Control Connection

- Targeted LDP session / BGP session / Static
  - Used for VC-label negotiation, withdrawal, error notification

The “emulated circuit” has three (3) layers of encapsulation

Tunnelling Component

- Tunnel header (Tunnel Label)
  - To get PDU from ingress to egress PE
  - MPLS LSP derived through static configuration (MPLS-TP) or dynamic (LDP or RSVP-TE)

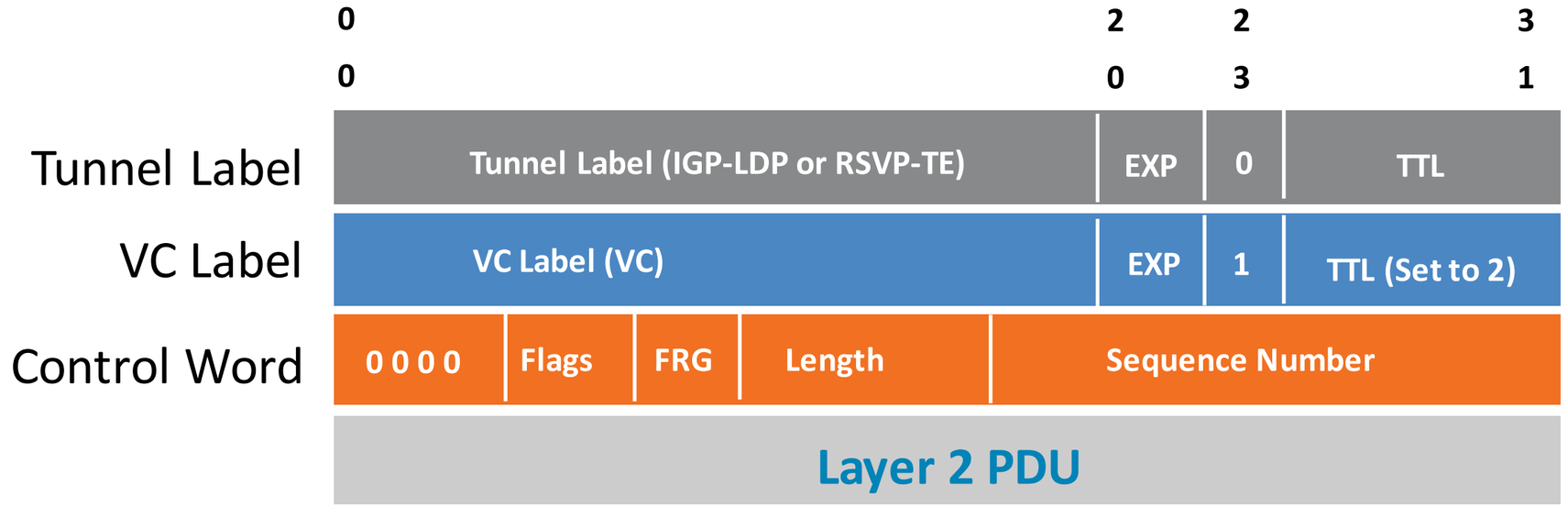
Demultiplexing Component

- Demultiplexer field (VC Label)
  - To identify individual circuits within a tunnel
  - Could be an MPLS label, L2TPv3 header, GRE key, etc.

Layer 2 Encapsulation

- Emulated VC encapsulation (Control Word)
  - Information on enclosed Layer 2 PDU
  - Implemented as a 32-bit control word

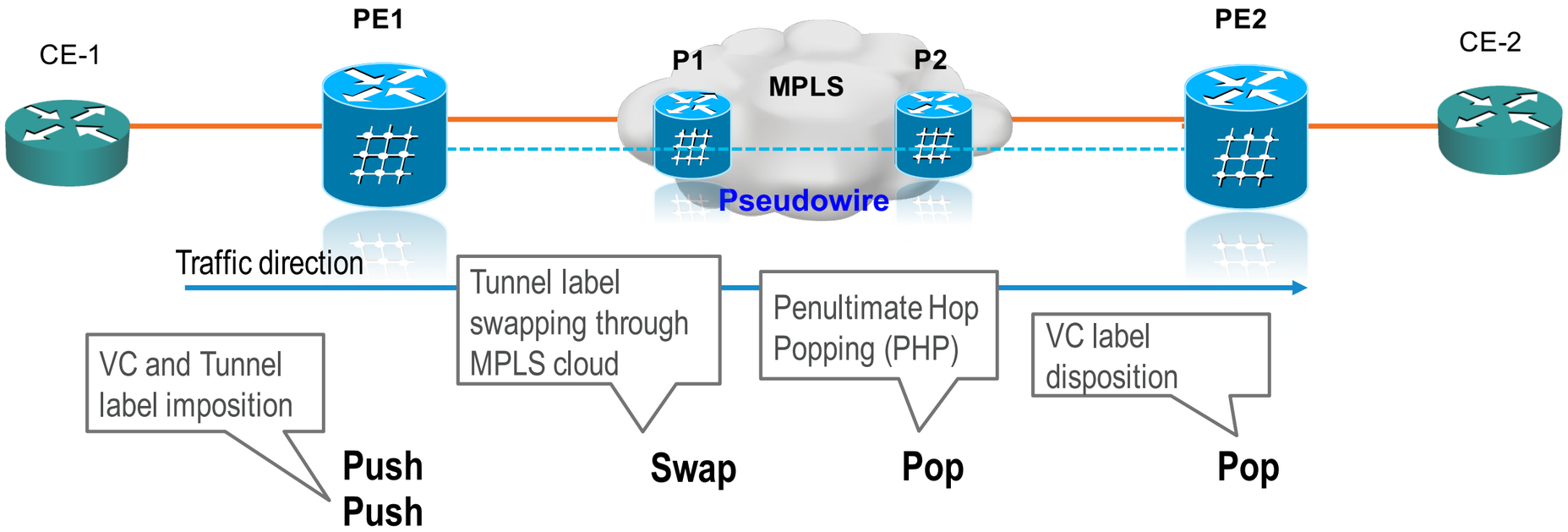
VPWS Traffic Encapsulation

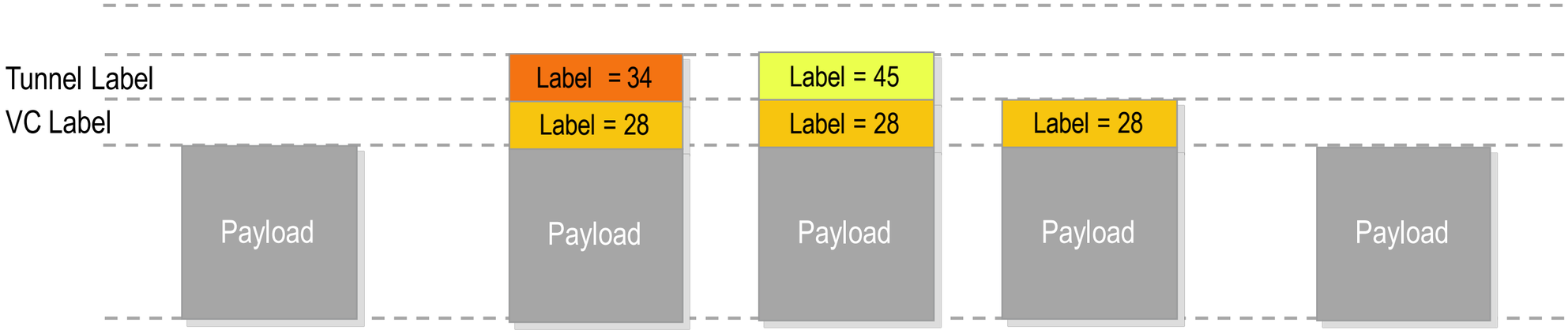


- Three-level encapsulation
- Packets switched between PEs using Tunnel label
- VC label identifies PW
- VC label signaled between PEs
- Optional Control Word (CW) carries Layer 2 control bits and enables sequencing

Control Word	
Encap.	Required
ATM N:1 Cell Relay	No
ATM AAL5	Yes
Ethernet	No
Frame Relay	Yes
HDLC	No
PPP	No
SAToP	Yes
CESoPSN	Yes

VPWS Forwarding Plane Processing



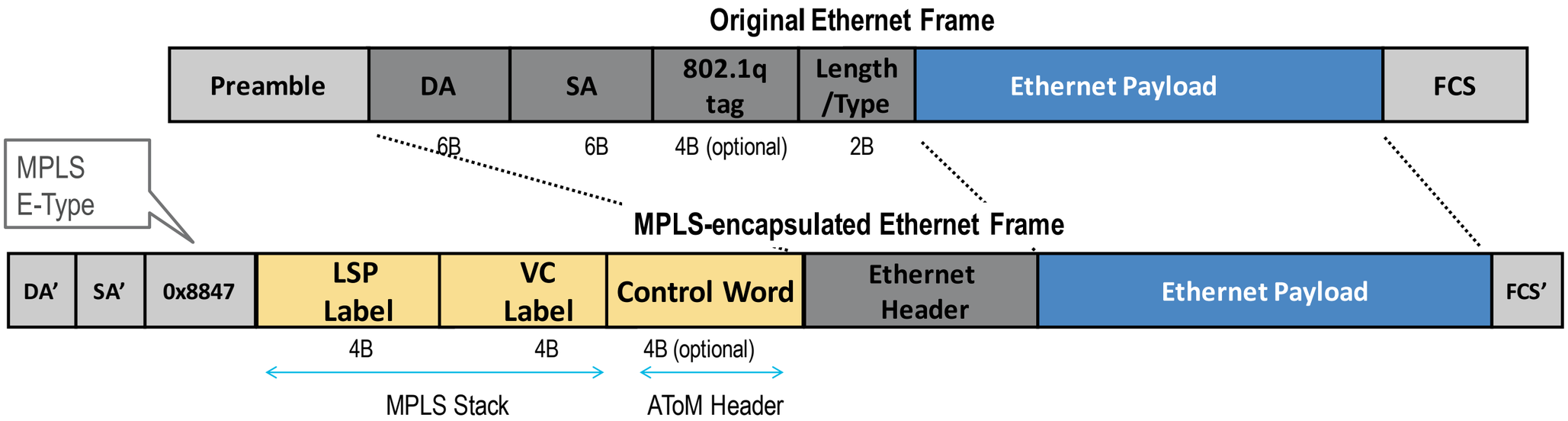


Virtual Private Wire Service (VPWS)

Ethernet over MPLS (EoMPLS)

How Are Ethernet Frames Transported?

- Ethernet frames transported without Preamble, Start Frame Delimiter (SFD) and FCS
- Two (2) modes of operation supported:
  - Ethernet VLAN mode (VC type 0x0004) – created for VLAN over MPLS application
  - Ethernet Port / Raw mode (VC type 0x0005) – created for Ethernet port tunneling application



Ethernet PW VC Type Negotiation

Cisco IOS

- Cisco devices by default will generally attempt to bring up an Ethernet PW using VC type 5
- If rejected by remote PE, then VC type 4 will be used
- Alternatively, Cisco device can be manually configured to use either VC type 4 or 5

```
7604-2(config-pw-class)#interworking ?
  ethernet  Ethernet interworking
  ip        IP interworking
  vlan      VLAN interworking

7604-2#show running-config
pseudowire-class test-pw-class-VC4
 encapsulation mpls
 interworking vlan
!
pseudowire-class test-pw-class-VC5
 encapsulation mpls
 interworking ethernet
```


Ethernet PW VC Type Negotiation

Cisco IOS-XR

- Cisco devices by default will generally attempt to bring up an Ethernet PW using VC type 5
- If rejected by remote PE, then VC type 4 will be used
- Alternatively, Cisco device can be manually configured to use either VC type 4 or 5

```
RP/0/RSP0/CPU0:ASR9000-2(config-l2vpn-pwc-
mpls)#transport-mode ?
  ethernet  Ethernet port mode
  vlan      Vlan tagged mode
RP/0/RSP0/CPU0:ASR9000-2(config-l2vpn-pwc-
mpls)#transport-mode vlan ?
  passthrough  passthrough incoming tags

RP/0/RSP0/CPU0:ASR9000-2#show running-config l2vpn
l2vpn
 pw-class test-pw-class-VC4
  encapsulation mpls
   transport-mode vlan

 pw-class test-pw-class-VC4-passthrough
  encapsulation mpls
   transport-mode vlan passthrough

 pw-class test-pw-class-VC5
  encapsulation mpls
   transport-mode ethernet
```


Introducing Cisco EVC Framework

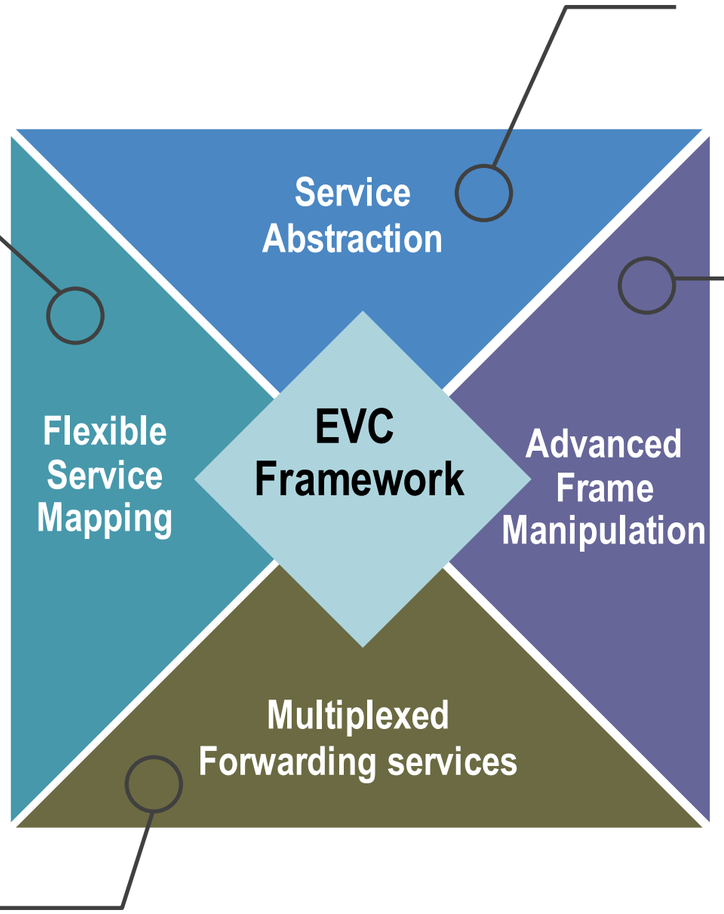
Functional Highlights

Ethernet Service Layer

- Ethernet Flow Point (EFP)
- Ethernet Virtual Circuit (EVC)
- Bridge Domain (BD)
- Local VLAN significance

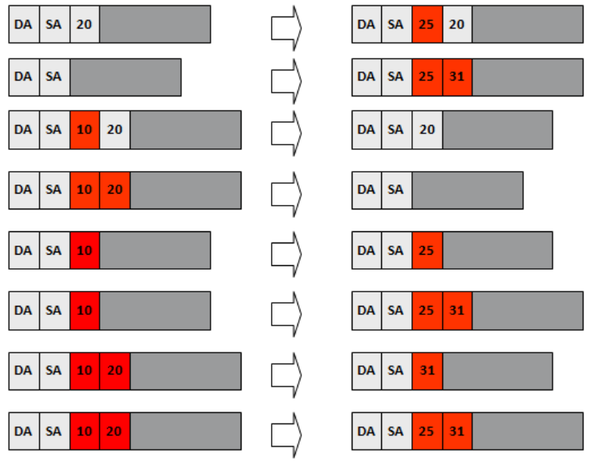
Flexible service delimiters

- Single-tagged, Double-tagged
- VLAN Lists, VLAN Ranges
- Header fields (COS, Ethertype)



VLAN Header operations - VLAN Rewrites

- POP
- PUSH
- SWAP



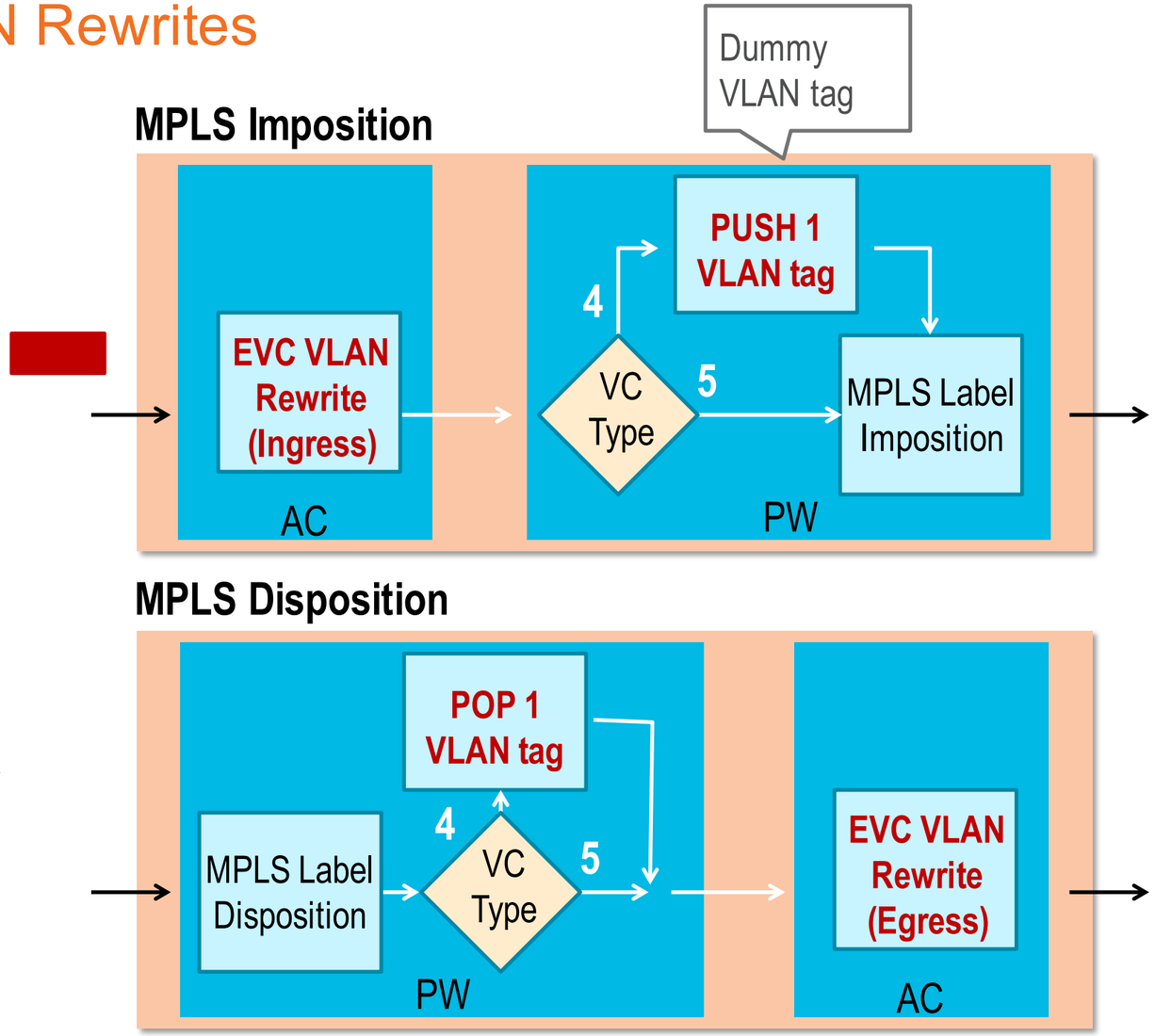
ANY service – ANY port

- Layer 2 Point-to-Point
- Layer 2 Multipoint
- Layer 3

Encapsulation Adjustment Considerations

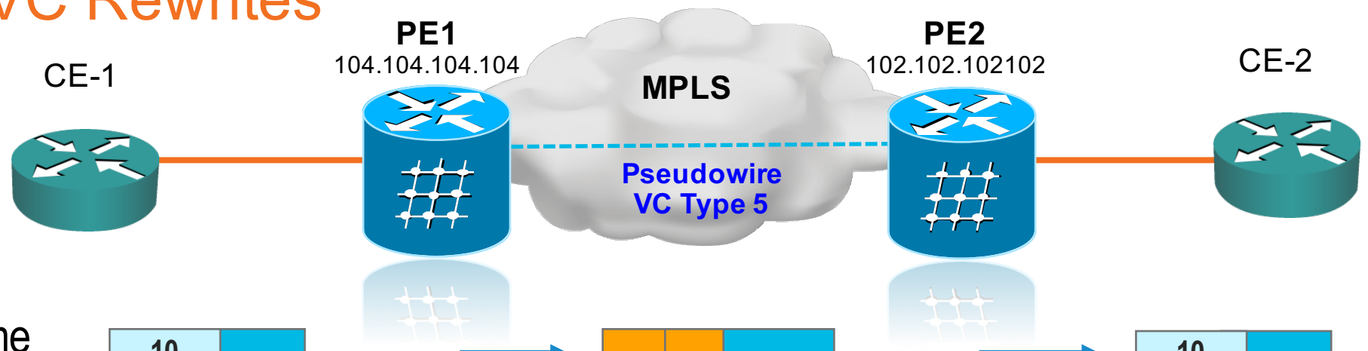
EoMPLS PW VC Type and EVC VLAN Rewrites

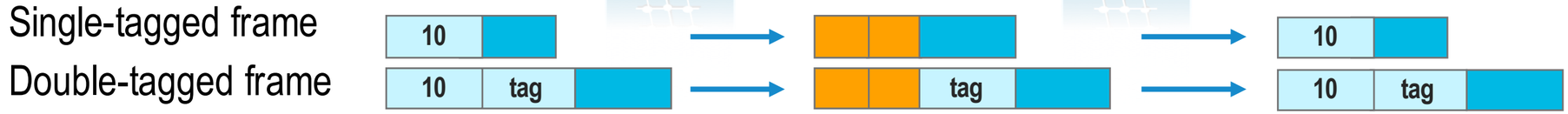
- VLAN tags can be added, removed or translated prior to VC label imposition or after disposition
  - Any VLAN tag(s), if retained, will appear as payload to the VC
- VC label imposition and service delimiting tag are independent from EVC VLAN tag operations
  - Dummy VLAN tag – RFC 4448 (sec 4.4.1)
- VC service-delimiting VLAN-ID is removed before passing packet to Attachment Circuit processing



Encapsulation Adjustment Considerations

VC 5 and EVC Rewrites





IOS-XR

```
l2vpn
 pw-class class-VC5
  encapsulation mpls
   transport-mode ethernet

 xconnect group Cisco-Live
  p2p xc-sample-1
   interface GigabitEthernet0/0/0/2.100
   neighbor 102.102.102.102 pw-id 111
    pw-class class-VC5
```

```
interface GigabitEthernet0/0/0/2.100 l2transport
 encapsulation dot1q 10
 rewrite ingress tag pop 1 symmetric
```

- POP VLAN 10
- No Push of Dummy tag (VC 5)

- No service-delimiting vlan expected (VC 5)
- PUSH VLAN 10

IOS

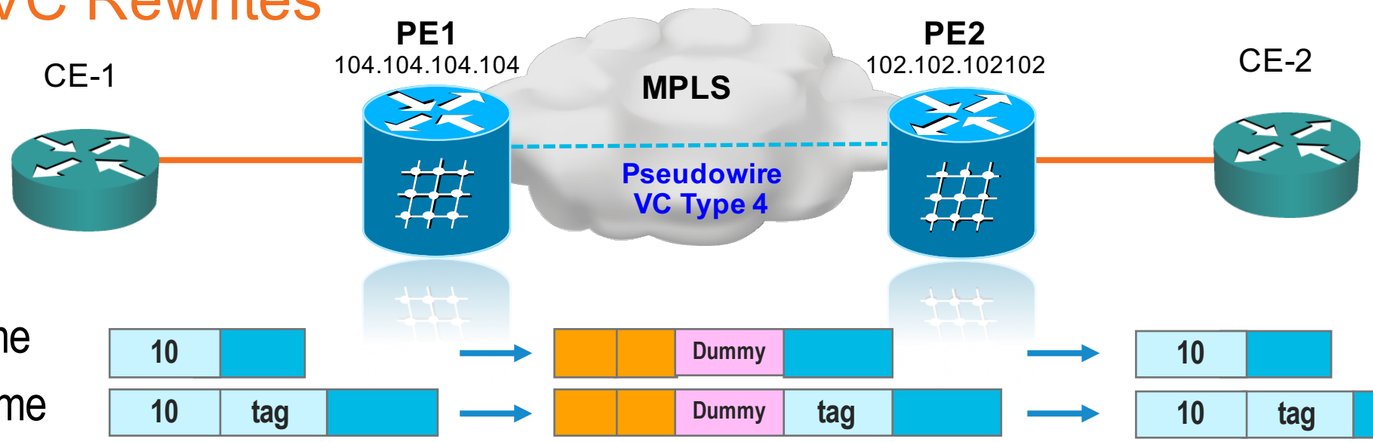
```
pseudowire-class class-VC5
 encapsulation mpls
 interworking ethernet
```

```
interface GigabitEthernet2/2
 service instance 3 ethernet
 encapsulation dot1q 10
 rewrite ingress tag pop 1 symmetric
 xconnect 104.104.104.104 111 encap mpls pw-class class-VC5
```

MPLS label

Encapsulation Adjustment Considerations

VC 4 and EVC Rewrites



IOS-XR

```
l2vpn
 pw-class class-VC4
  encapsulation mpls
   transport-mode vlan

 xconnect group Cisco-Live
  p2p xc-sample-1
   interface GigabitEthernet0/0/0/2.100
   neighbor 102.102.102.102 pw-id 111
    pw-class class-VC4
```

```
interface GigabitEthernet0/0/0/2.100 l2transport
 encapsulation dot1q 10
 rewrite ingress tag pop 1 symmetric
```

- POP VLAN 10
- Push Dummy tag (VC 4)

- POP service-delimiting vlan (VC 4)
- PUSH VLAN 10

IOS

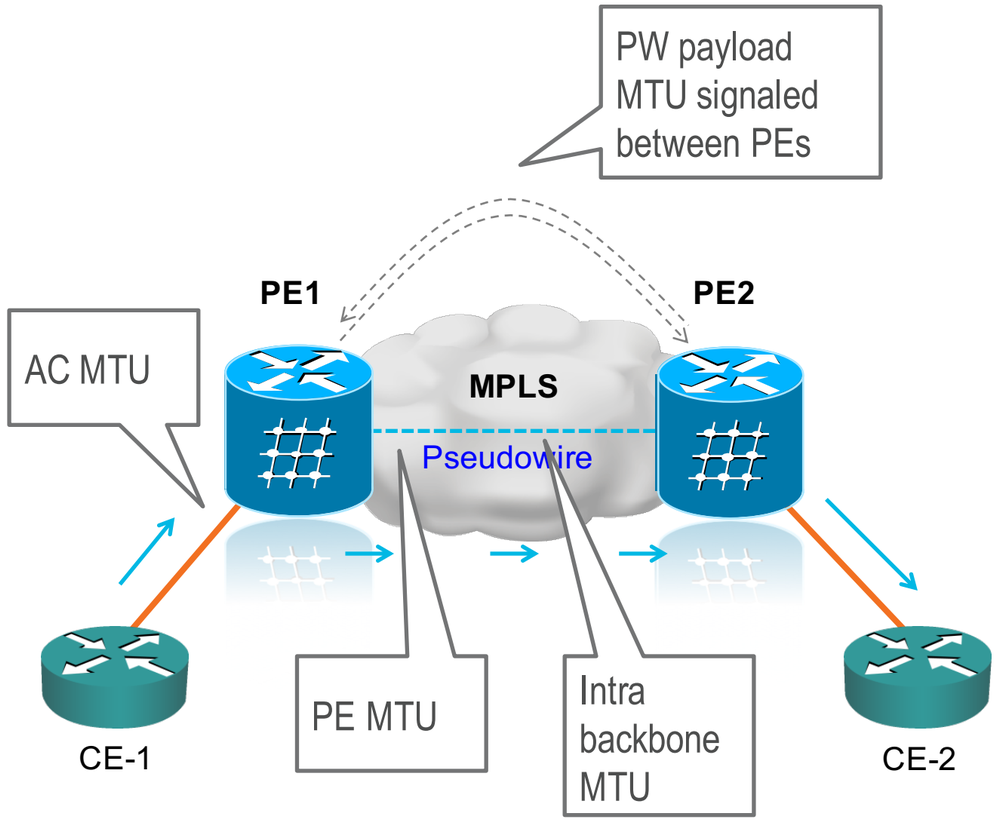
```
pseudowire-class class-VC4
 encapsulation mpls
 interworking vlan
```

```
interface GigabitEthernet2/2
 service instance 3 ethernet
 encapsulation dot1q 10
 rewrite ingress tag pop 1 symmetric
 xconnect 104.104.104.104 111 encap mpls pw-class class-VC4
```

MPLS label

MTU Considerations

- No payload fragmentation supported
- Incoming PDU dropped if MTU exceeds AC MTU
- PEs exchange PW payload MTU as part of PW signaling procedures
  - Both ends must agree to use same value for PW to come UP
  - PW MTU derived from AC MTU
- No mechanism to check Backbone MTU
  - MTU in the backbone must be large enough to carry PW payload and MPLS stack



Ethernet MTU Considerations

Cisco IOS

- Interface MTU configured as largest ethernet payload size
  - 1500B default
  - Sub-interfaces / Service Instances (EFPs) MTU always inherited from main interface
- PW MTU used during PW signaling
  - By default, inherited from attachment circuit MTU
  - Submode configuration CLI allows MTU value to be set per subinterface/EFP in xconnect configuration mode (only for signaling purposes)
  - No MTU adjustments made for EFP rewrite (POP/PUSH) operations

```
interface GigabitEthernet0/0/4
 description Main interface
 mtu 1600
```

```
ASR1004-1#show int gigabitEthernet 0/0/4.1000 | include MTU
  MTU 1600 bytes, BW 100000 Kbit/sec, DLY 100 usec,
```

Sub-interface MTU inherited from Main interface

```
interface GigabitEthernet0/0/4.1000
 encapsulation dot1Q 1000
 xconnect 106.106.106.106 111 encapsulation mpls
  mtu 1500
```

PW MTU used during signaling can be overwritten

Ethernet MTU Considerations

Cisco IOS XR

- Interface / sub-interface MTU configured as largest frame size – FCS (4B)
  - 1514B default for main interfaces
  - 1518B default for single-tagged subinterfaces
  - 1522B default for double-tagged subinterfaces
- PW MTU used during PW signaling
  - AC MTU – 14B + Rewrite offset
  - E.g. POP 1 (- 4B), PUSH 1 (+ 4B)

```
interface GigabitEthernet0/0/0/2
 description Main interface
 mtu 9000
```

```
interface GigabitEthernet0/0/0/2.100 l2transport
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric
 mtu 1518
```

By default, sub-interface MTU inherited from Main interface

Sub-interface MTU can be overwritten to match remote AC

```
RP/0/RSP0/CPU0:PE1#show l2vpn xconnect neighbor 102.102.102.102 pw-
id 11
Group Cisco-Live, XC xc-sample-1, state is down; Interworking none
  AC: GigabitEthernet0/0/0/2.100, state is up
    Type VLAN; Num Ranges: 1
    VLAN ranges: [100, 100]
    MTU 1500; XC ID 0x840014; interworking none
    Statistics:
(snip)
```

XC MTU = 1518 – 14 – 4 = 1500B

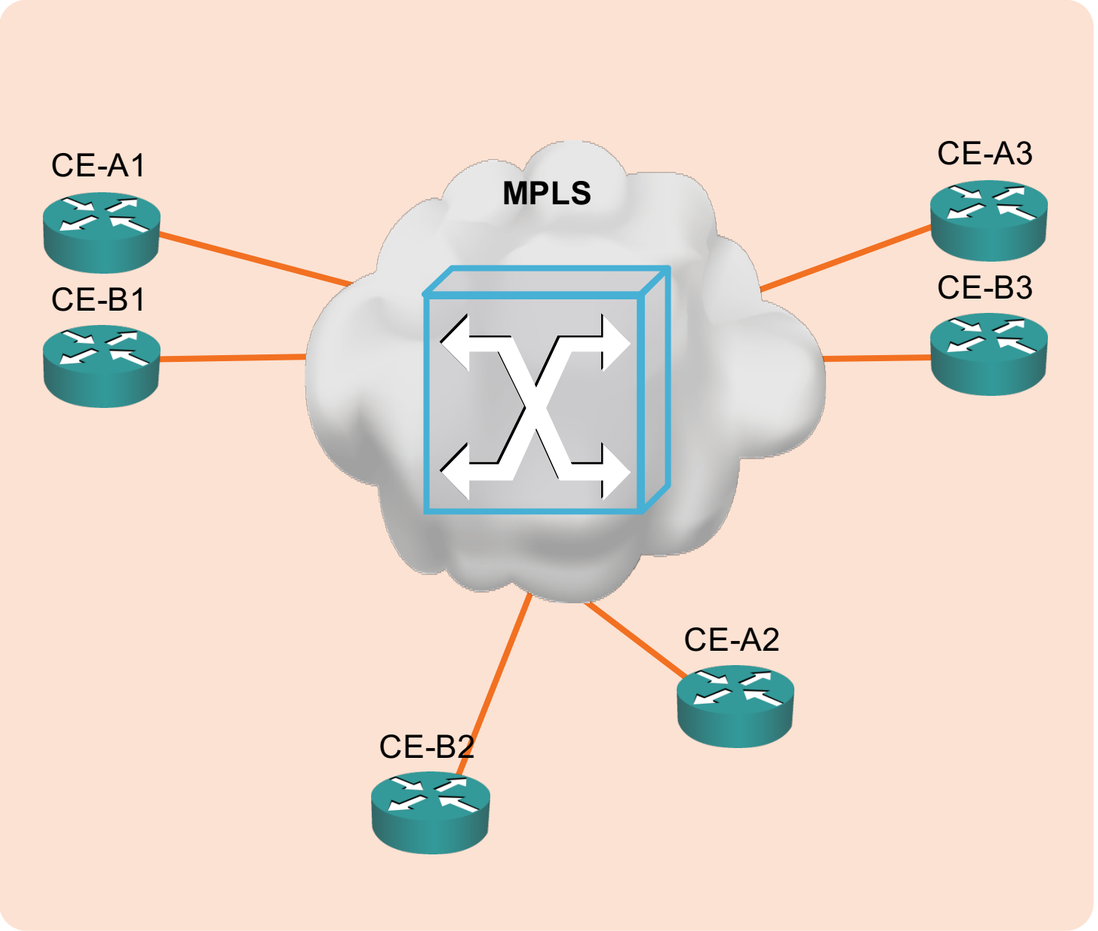
Virtual Private LAN Service (VPLS)

Overview

Virtual Private LAN Service

Overview

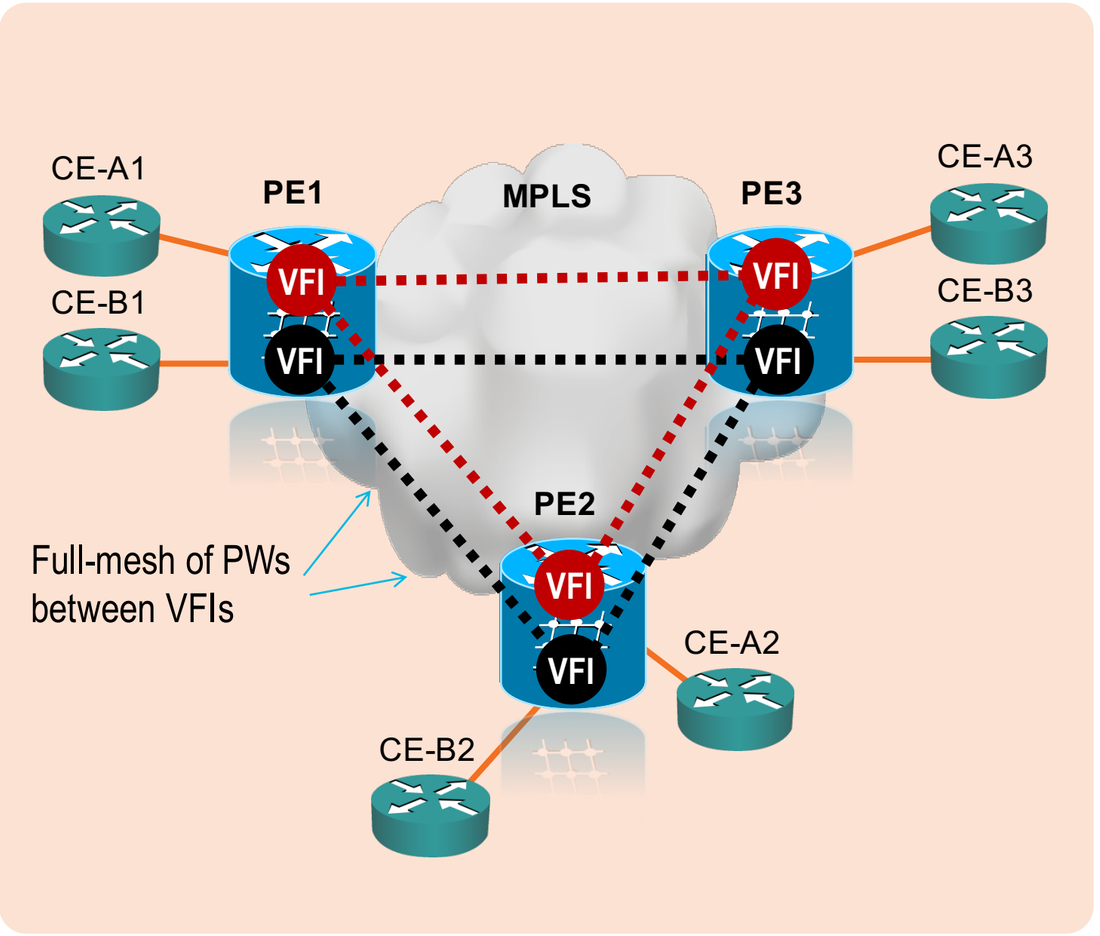
- Defines Architecture to provide Ethernet Multipoint connectivity sites, as if they were connected using a LAN
- VPLS operation emulates an IEEE Ethernet switch
- Two (2) signaling methods
  - RFC 4762 (LDP-Based VPLS)
  - RFC 4761 (BGP-Based VPLS)



Virtual Private LAN Service

Reference Model

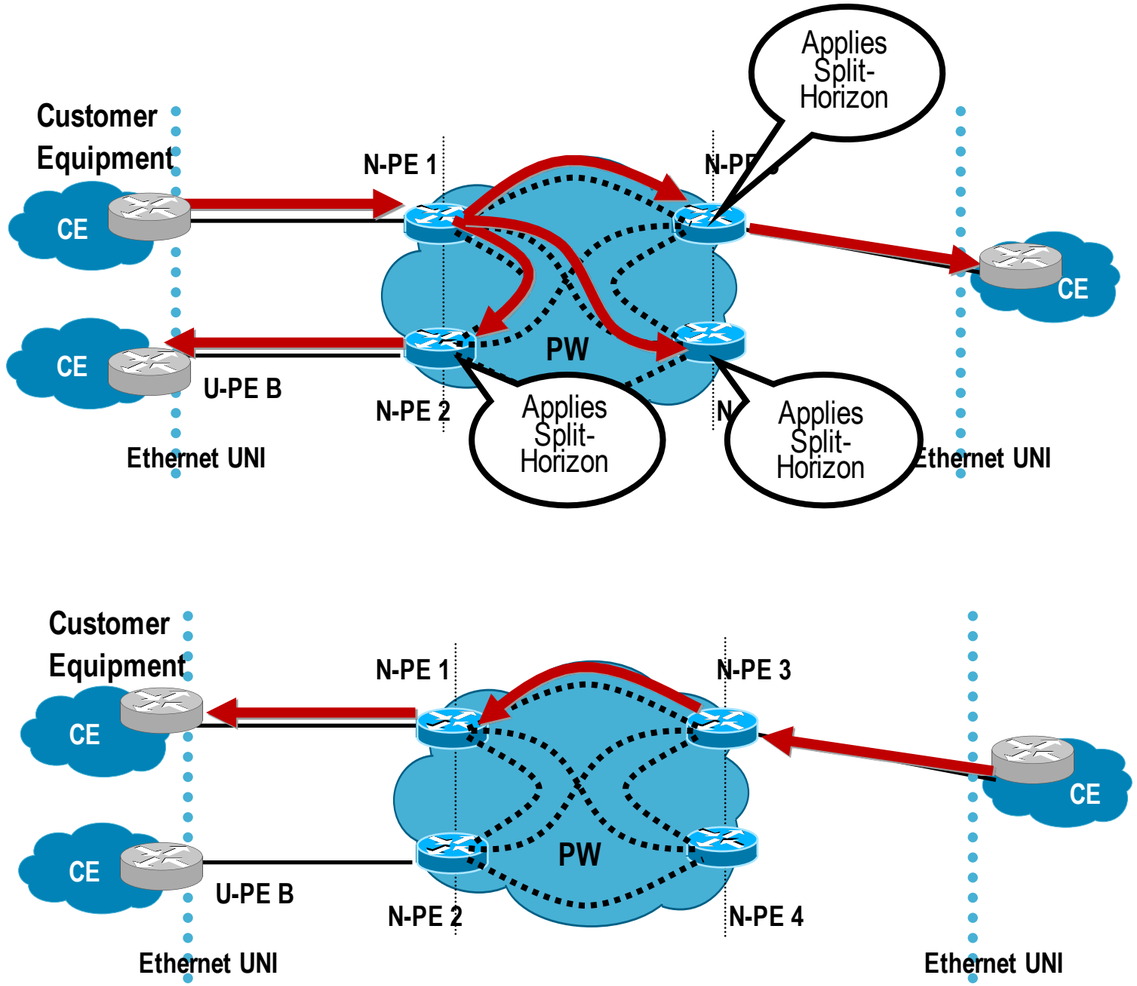
- VFI (Virtual Forwarding Instance)
  - Also called VSI (Virtual Switching Instance)
  - Emulates L2 broadcast domain among ACs and VCs
  - Unique per service. Multiple VFIs can exist same PE
- AC (Attachment Circuit)
  - Connect to CE device, it could be Ethernet physical or logical port
  - One or multiple ACs can belong to same VFI
- VC (Virtual Circuit)
  - EoMPLS data encapsulation, tunnel label used to reach remote PE, VC label used to identify VFI
  - One or multiple VCs can belong to same VFI
  - PEs must have a full-mesh of PWs in the VPLS core



Virtual Private LAN Service

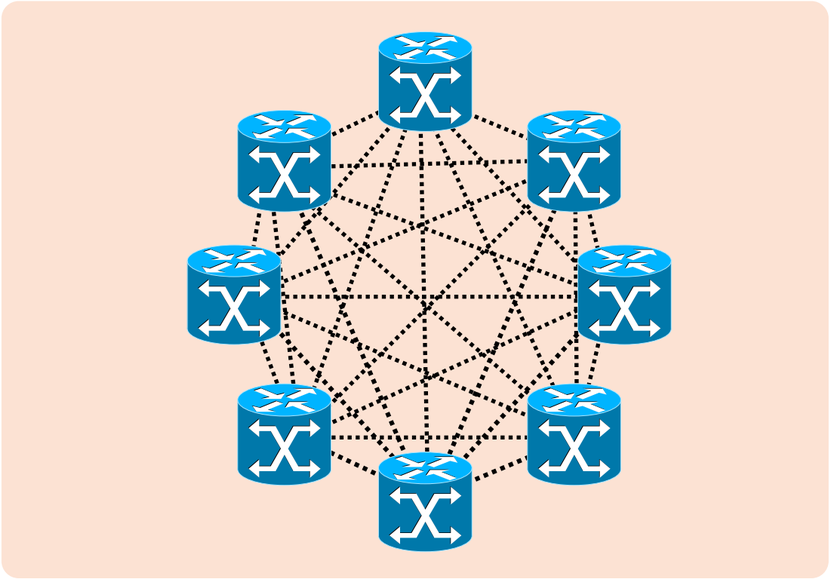
Operation

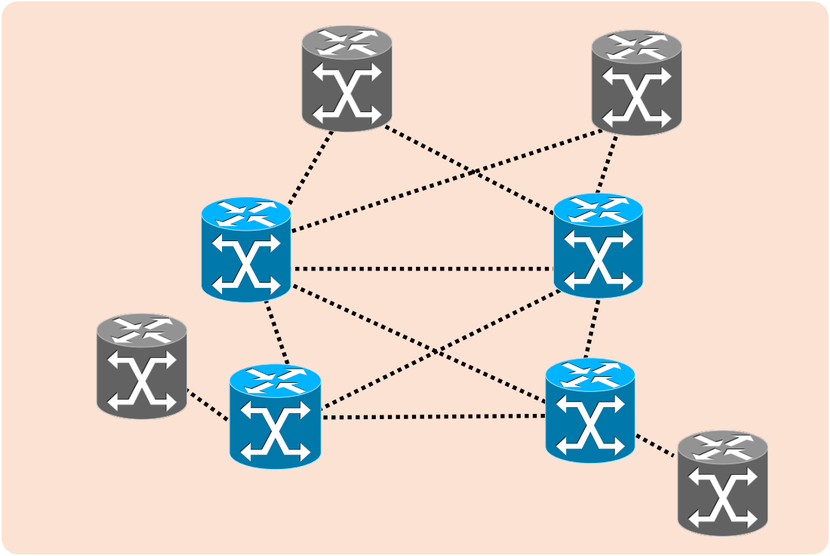
- Flooding / Forwarding
  - Forwarding based on destination MAC addresses
  - Flooding (Broadcast, Multicast, Unknown Unicast)
- MAC Learning/Aging/Withdrawal
  - Dynamic learning based on Source MAC and VLAN
  - Refresh aging timers with incoming packet
  - MAC withdrawal upon topology changes
- Split-Horizon and Full-Mesh of PWs for loop-avoidance in core
  - SP does not run STP in the core



Why H-VPLS? Improved Scaling

- Flat VPLS
  - Potential signaling overhead
  - Packet replication at the edge
  - Full PW mesh end-end
- Hierarchical-VPLS
  - Minimizes signaling overhead
  - Packet replication at the core only
  - Full PW mesh in the core

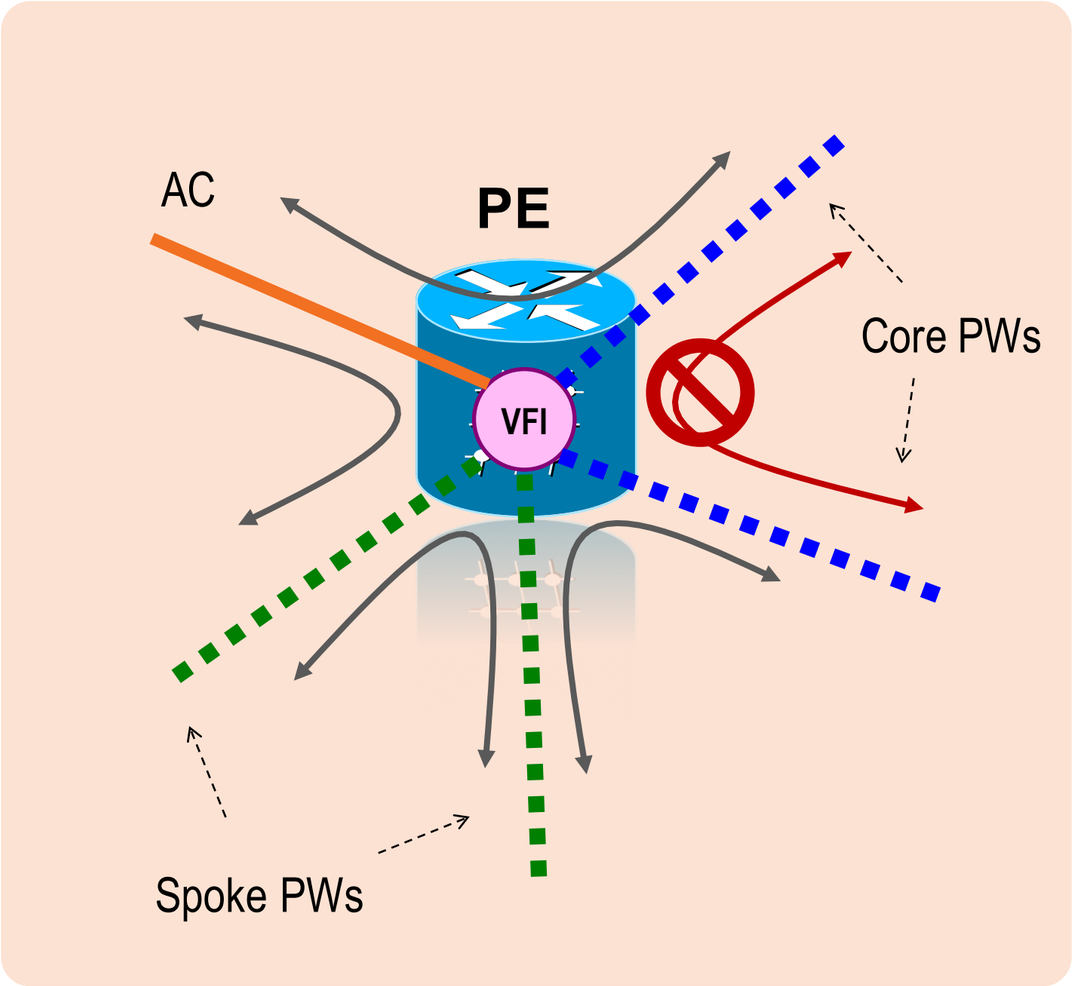




VPLS Operation

Loop Prevention

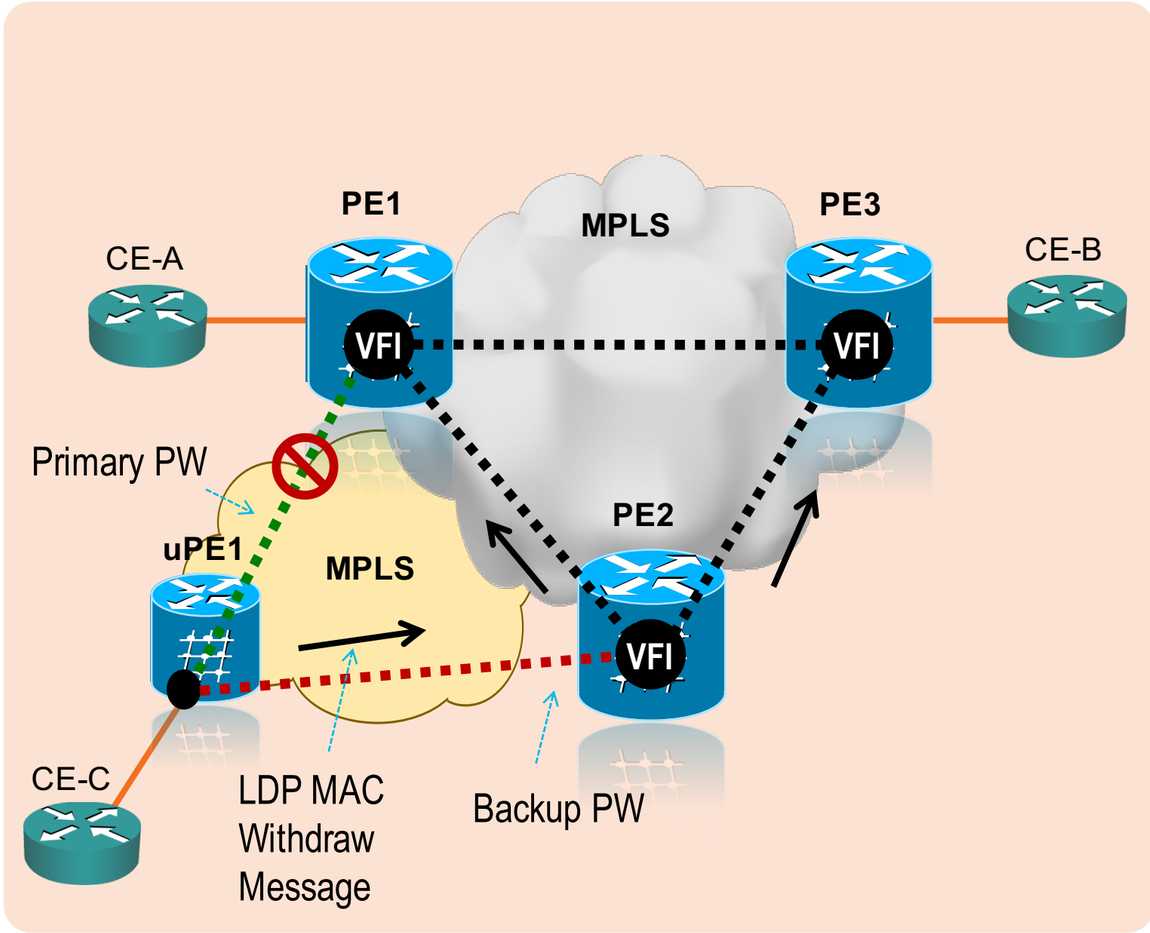
- Core PW – Split Horizon ON
- Spoke PW – Split Horizon OFF (default)
- Split-Horizon Rules
  - Forwarding between Spoke PWs
  - Forwarding between Spoke and Core PWs
  - Forwarding between ACs and Core / Spoke PWs
  - Forwarding between ACs
  - Blocking between Core PWs



VPLS Operation

MAC Address Withdrawal

- Remove (flush) dynamic MAC addresses upon Topology Changes
  - Faster convergence – avoids blackholing
  - Uses LDP Address Withdraw Message (RFC 4762)
- H-VPLS dual-home example
  - U-PE detects failure of Primary PW
  - U-PE activates Backup PW
  - U-PE sends LDP MAC address withdrawal request to new N-PE
  - N-PE forwards the message to all PWs in the VPLS core and flush its MAC address table

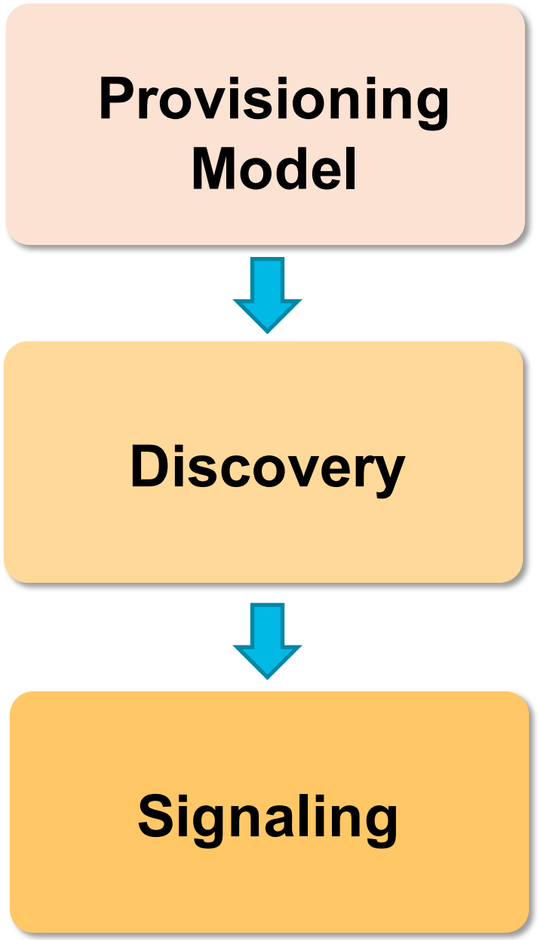


Pseudowire (PW) Signaling and PE Auto-Discovery

VPWS / VPLS

An abstraction

- Provisioning Model
  - What information needs to be configured and in what entities
  - Semantic structure of the endpoint identifiers (e.g. VC ID, VPN ID)
- Discovery
  - Provisioning information is distributed by a "discovery process“
  - Distribution of endpoint identifiers
- Signaling
  - When the discovery process is complete, a signaling protocol is automatically invoked to set up pseudowires (PWs)

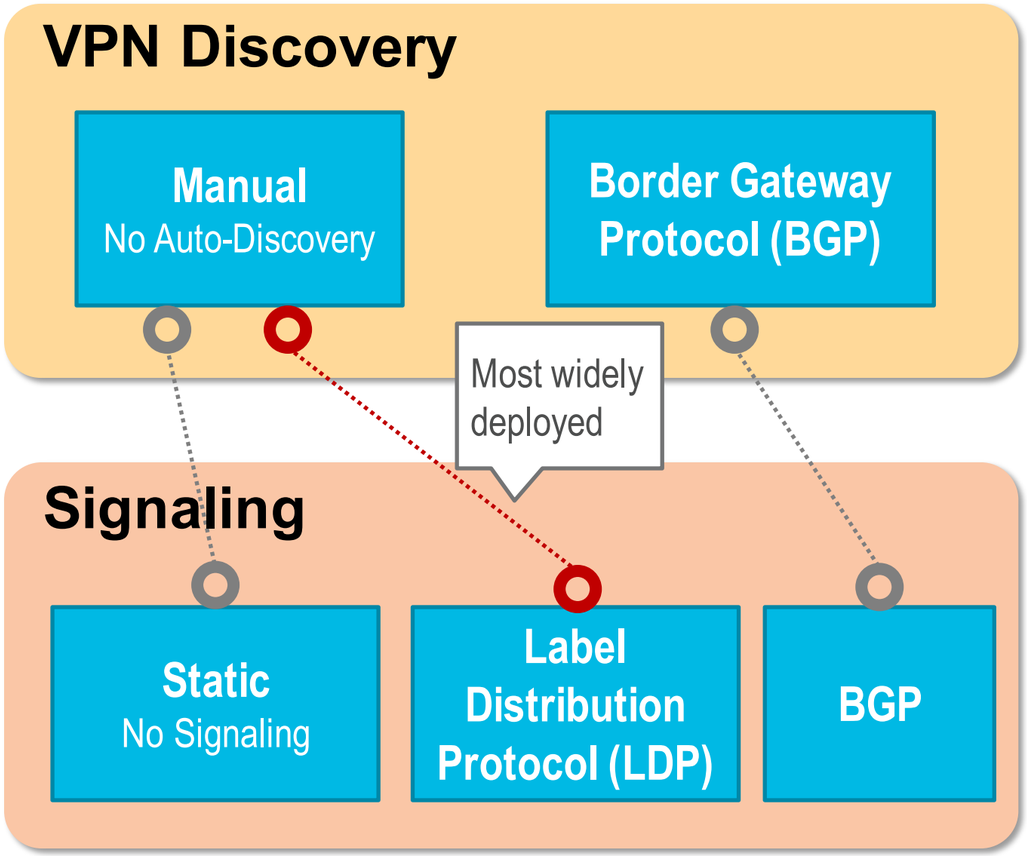


VPWS

Discovery and Signaling Alternatives

- VPWS Signaling
  - LDP-based (RFC 4447)
  - BGP-based (informational draft)

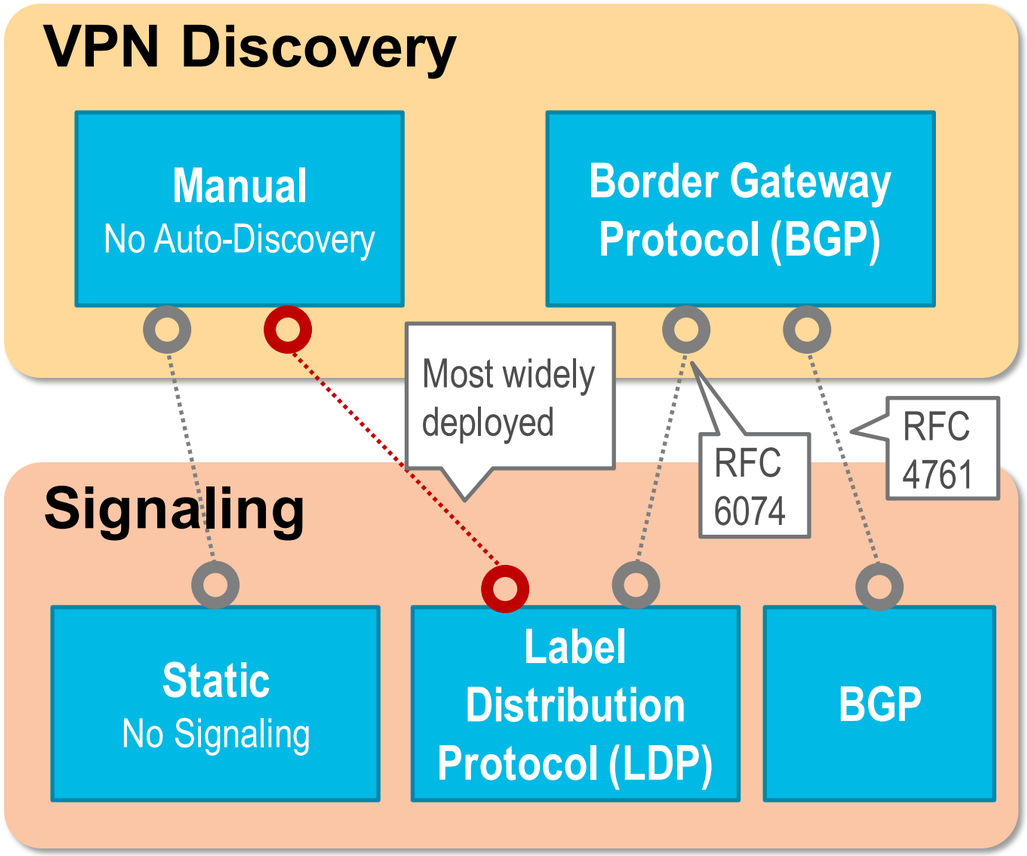
    draft-kompella-l2vpn-l2vpn
- VPWS with LDP-signaling and No auto-discovery
  - Most widely deployed solution
- Auto-discovery for point-to-point services not as relevant as for multipoint



VPLS

Discovery and Signaling Alternatives

- VPLS Signaling
  - LDP-based (RFC 4762)
  - BGP-based (RFC 4761)
- VPLS with LDP-signaling and No auto-discovery
  - Most widely deployed solution
  - Operational complexity for larger deployments
- BGP-based Auto-Discovery (BGP-AD) (RFC 6074)
  - Enables discovery of PE devices in a VPLS instance

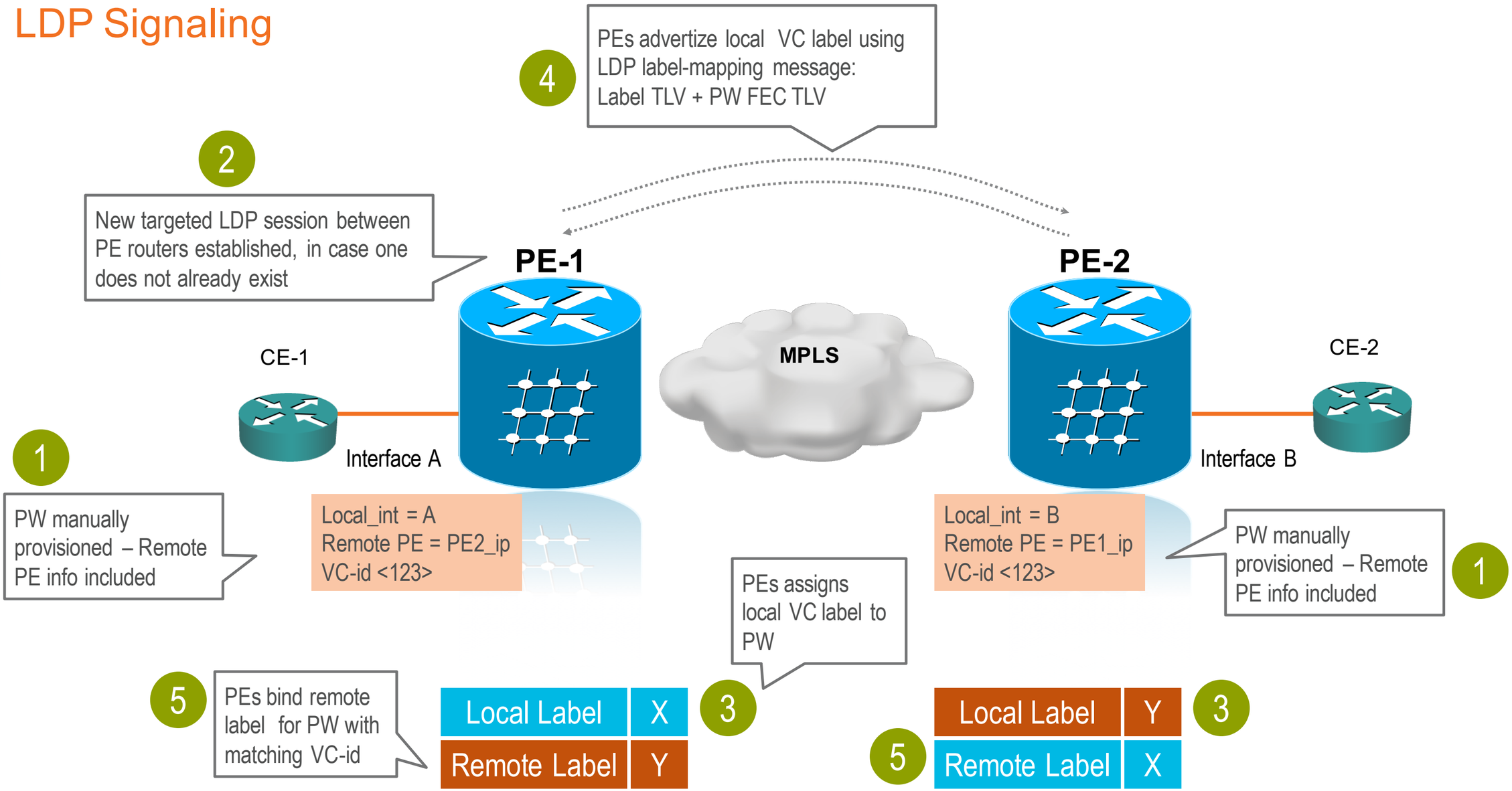


Pseudowire (PW) Signaling and PE Auto-Discovery

LDP-based Signaling and Manual Provisioning

PW Control Plane Operation

LDP Signaling



VPWS (EoMPLS) LDP Signaling

Cisco IOS (VLAN-based services)

```
hostname PE1
!
interface Loopback0
 ip address 106.106.106.106 255.255.255.255
```

```
interface GigabitEthernet2/4.300
 encapsulation dot1q 300
 xconnect 102.102.102.102 111 encapsulation mpls
```

Sub-interface based xconnect

OR

```
interface GigabitEthernet2/4
 service instance 10 ethernet
  encapsulation dot1q 300
  rewrite ingress tag pop 1 symmetric
  xconnect 102.102.102.102 111 encapsulation mpls
```

Service-Instance (EFP) based xconnect

OR

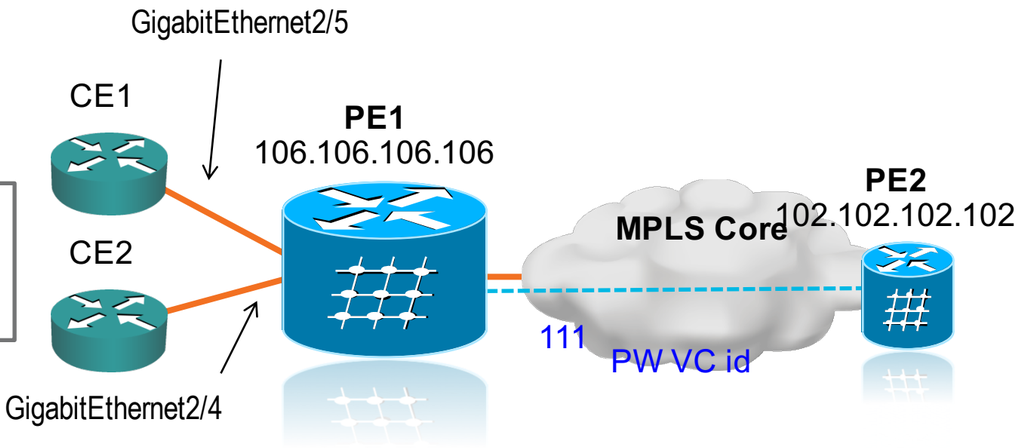
```
interface Vlan 300
 xconnect 102.102.102.102 111 encapsulation mpls
!
interface GigabitEthernet2/4
 switchport mode trunk
 switchport trunk allowed vlan 300
```

Interface VLAN (SVI) based xconnect + Switchport trunk / access

OR

```
interface Vlan 300
 xconnect 102.102.102.102 111 encapsulation mpls
!
interface GigabitEthernet2/4
 service instance 10 ethernet
  encapsulation dot1q 300
  rewrite ingress tag pop 1 symmetric
  bridge-domain 300
```

Interface VLAN (SVI) based xconnect + Service instance BD



VPWS (EoMPLS) LDP Signaling

Cisco IOS (Port-based services)

```
hostname PE1
!
interface Loopback0
 ip address 106.106.106.106 255.255.255.255
```

Main interface based xconnect

```
interface GigabitEthernet2/5
 xconnect 102.102.102.102 222 encapsulation mpls
```

OR

```
interface GigabitEthernet2/5
 service instance 1 ethernet
  encapsulation default
  xconnect 102.102.102.102 111 encapsulation mpls
```

Service-Instance (EFP) based xconnect (encap default)

OR

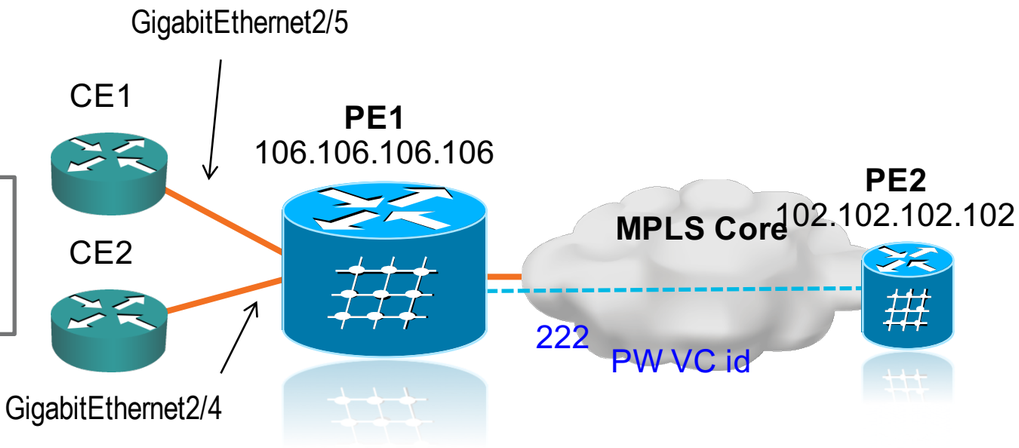
```
interface Vlan 300
 xconnect 102.102.102.102 111 encapsulation mpls
!
interface GigabitEthernet2/5
 switchport mode dot1q-tunnel
 switchport access vlan 300
```

Interface VLAN (SVI) based xconnect + Switchport dot1q-tunnel

OR

```
interface Vlan 300
 xconnect 102.102.102.102 111 encapsulation mpls
!
interface GigabitEthernet2/5
 service instance 1 ethernet
  encapsulation default
  bridge-domain 300
```

Interface VLAN (SVI) based xconnect + Service instance BD



VPWS (EoMPLS) LDP Signaling

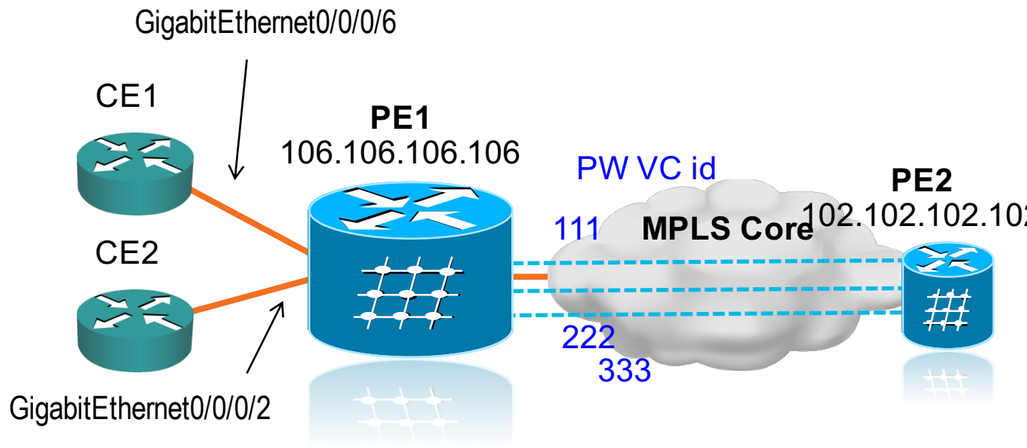
Cisco IOS XR

```
hostname PE1
!
interface Loopback0
 ipv4 address 106.106.106.106 255.255.255.255
```

```
l2vpn
 xconnect group Cisco-Live
  p2p xc-sample-1
   interface GigabitEthernet0/0/0/2.100
   neighbor 102.102.102.102 pw-id 111

  p2p xc-sample-2
   interface GigabitEthernet0/0/0/2.200
   neighbor 102.102.102.102 pw-id 222

  p2p xc-sample-3
   interface GigabitEthernet0/0/0/6
   neighbor 102.102.102.102 pw-id 333
```



Single-tagged VLAN traffic to PW

```
interface GigabitEthernet0/0/0/2.100 l2transport
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric
```

Single-tagged range VLAN traffic to PW

```
interface GigabitEthernet0/0/0/2.200 l2transport
 encapsulation dot1q 999-1010
 rewrite ingress tag push dot1q 888 symmetric
```

OR

Entire port traffic to PW

```
interface GigabitEthernet0/0/0/6
 l2transport
```


VPLS LDP Signaling / Manual provisioning

Cisco IOS

```
hostname PE1
!
interface Loopback0
 ip address 192.0.0.1 255.255.255.255
!
l2 vfi sample-vfi manual
 vpn id 1111
 neighbor 192.0.0.2 1111 encapsulation mpls
 neighbor 192.0.0.3 2222 encapsulation mpls
 neighbor 192.0.0.4 3333 encapsulation mpls
!
interface Vlan300
 xconnect vfi sample-vfi
```

VPN ID defined per VFI or on a per-neighbor basis

Core PWs Full-mesh

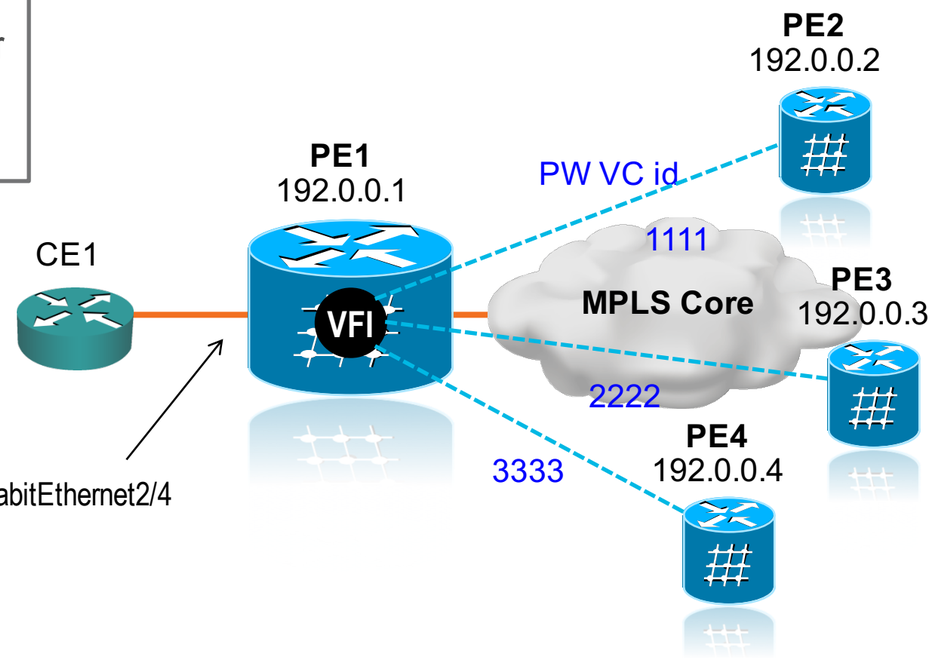
VFI associated to VLAN interface (SVI) via xconnect cmd

Bridge-Domain or VLAN/switchport configurations

```
interface GigabitEthernet2/4
 service instance 333 ethernet
  encapsulation dot1q 333
  rewrite ingress tag pop 1 symmetric
```

OR

```
interface GigabitEthernet2/4
 switchport mode trunk
```

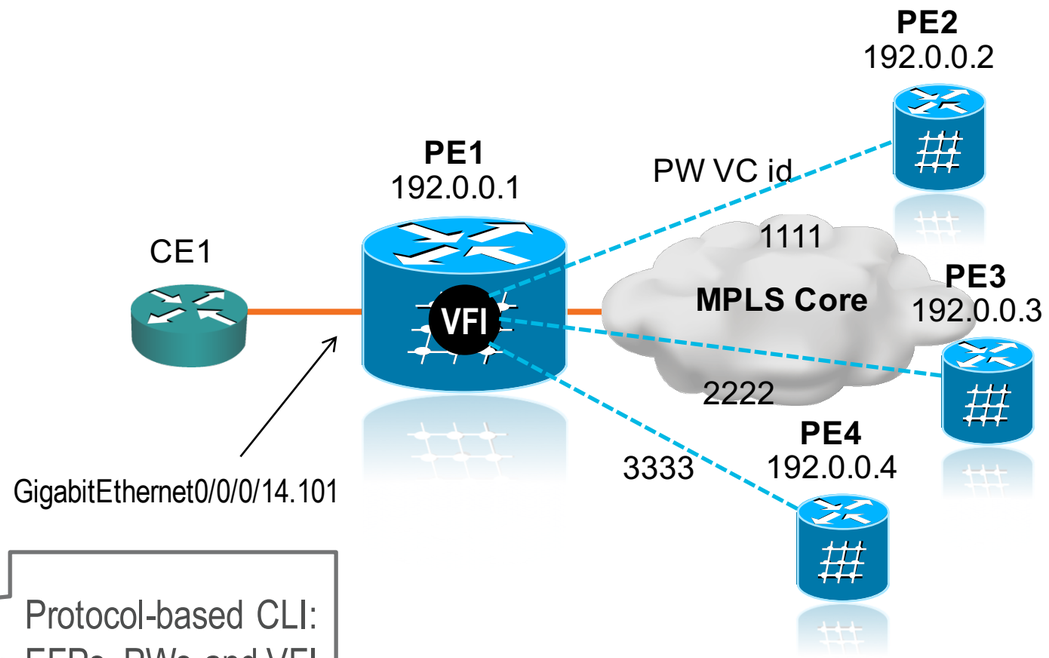


VPLS LDP Signaling / Manual provisioning

Cisco IOS XR

```
hostname PE1
!
interface Loopback0
 ipv4 address 192.0.0.1 255.255.255.255
!
interface GigabitEthernet0/0/0/14.101 l2transport
 encapsulation dot1q 101
 rewrite ingress tag pop 1 symmetric
```

```
l2vpn
 bridge group Cisco-Live
  bridge-domain bd101
   interface GigabitEthernet0/0/0/14.101
   vfi vfi101
    vpn-id 1111
    neighbor 192.0.0.2 pw-id 1111
    neighbor 192.0.0.3 pw-id 2222
    neighbor 192.0.0.4 pw-id 3333
```



Protocol-based CLI: EFPs, PWs and VFI as members of Bridge Domain

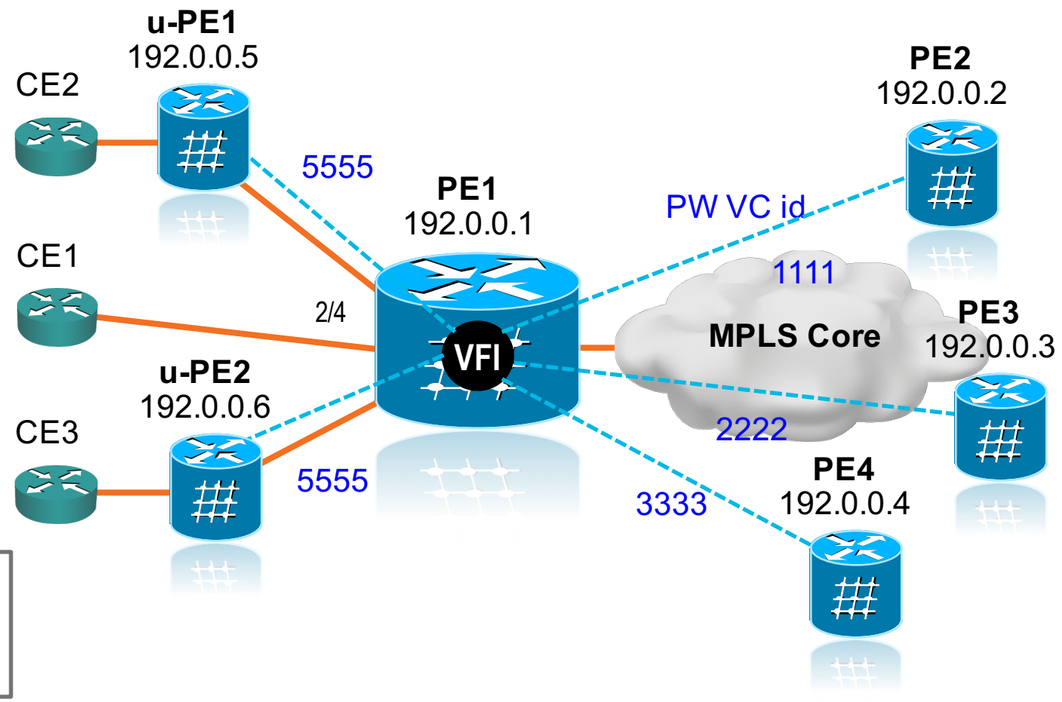
VPN ID defined per VFI or on a per-neighbor basis

H-VPLS LDP Signaling / Manual provisioning

Cisco IOS

```
hostname PE1
!
interface Loopback0
 ip address 192.0.0.1 255.255.255.255
!
l2 vfi sample-vfi manual
 vpn id 1111
 neighbor 192.0.0.2 encapsulation mpls
 neighbor 192.0.0.3 2222 encapsulation mpls
 neighbor 192.0.0.4 3333 encapsulation mpls
 neighbor 192.0.0.5 5555 encapsulation mpls no-split-horizon
 neighbor 192.0.0.6 5555 encapsulation mpls no-split-horizon
!
interface Vlan300
 xconnect vfi sample-vfi
```

Spoke PWs



Bridge-Domain or VLAN/switchport configurations

```
interface GigabitEthernet2/4
 service instance 333 ethernet
  encapsulation dot1q 333
  rewrite ingress tag pop 1 symmetric
```

OR

```
interface GigabitEthernet2/4
 switchport mode trunk
```


H-VPLS LDP Signaling / Manual provisioning

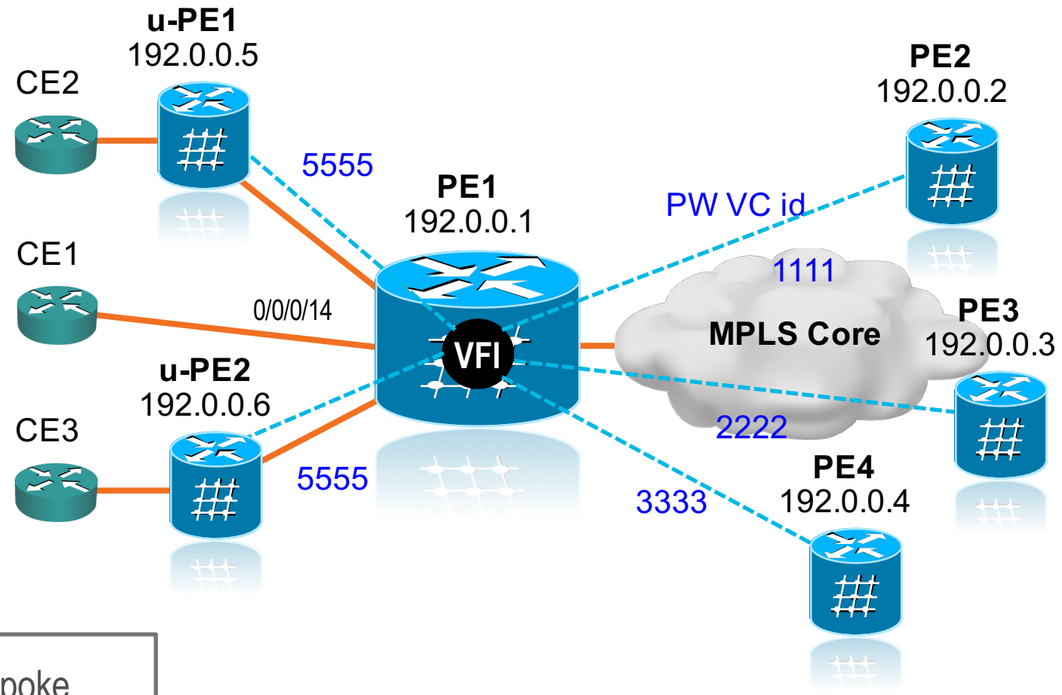
Cisco IOS XR

```
hostname PE1
!
interface Loopback0
 ipv4 address 192.0.0.1 255.255.255.255
!
interface GigabitEthernet0/0/0/14.101 l2transport
 encapsulation dot1q 101
 rewrite ingress tag pop 1 symmetric
```

```
l2vpn
 bridge group Cisco-Live
  bridge-domain bd101
   interface GigabitEthernet0/0/0/14.101
   neighbor 192.0.0.5 pw-id 5555
   neighbor 192.0.0.6 pw-id 5555
   !
   vfi vfi101
    vpn-id 1111
    neighbor 192.0.0.2 pw-id 1111
    neighbor 192.0.0.3 pw-id 2222
    neighbor 192.0.0.4 pw-id 3333
```

Spoke PWs

Core PWs Full-mesh

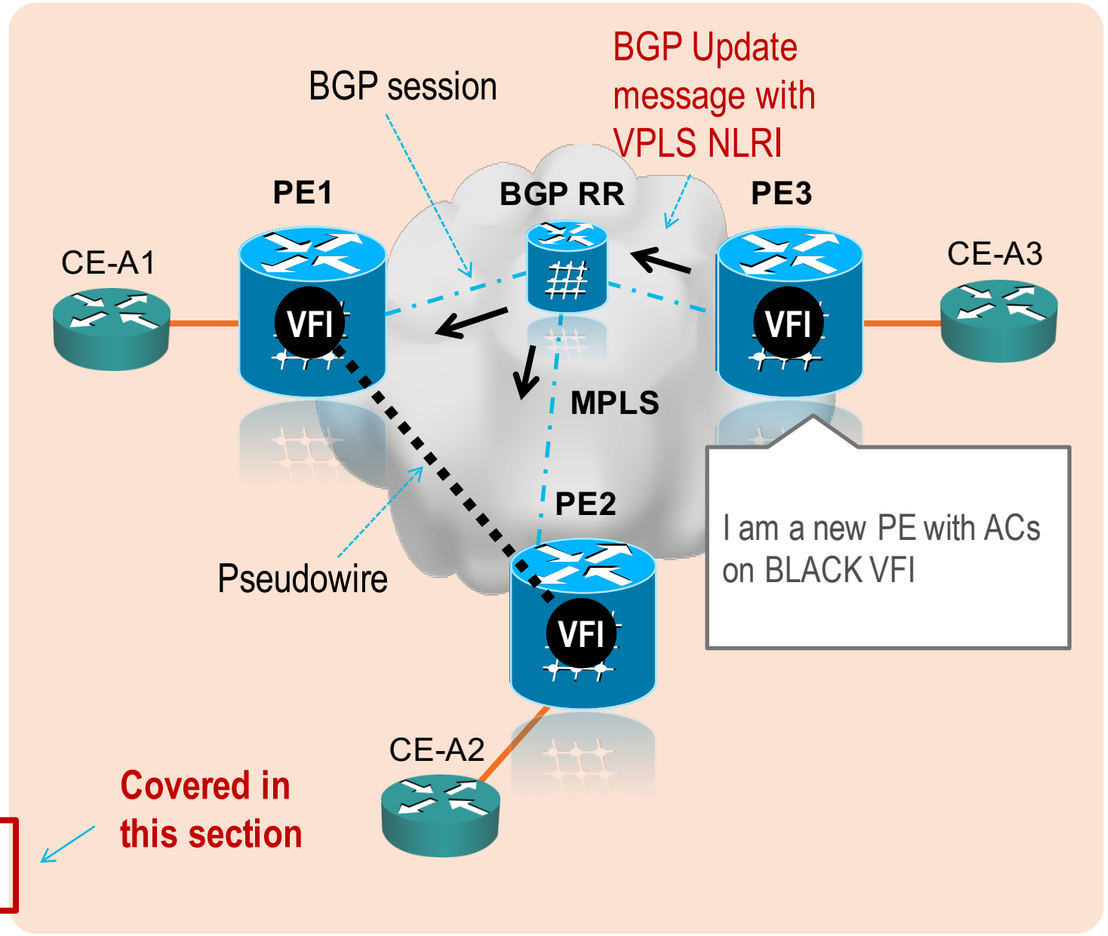


Pseudowire (PW) Signaling and PE Auto-Discovery

BGP-based AutoDiscovery (BGP-AD) and LDP Signaling

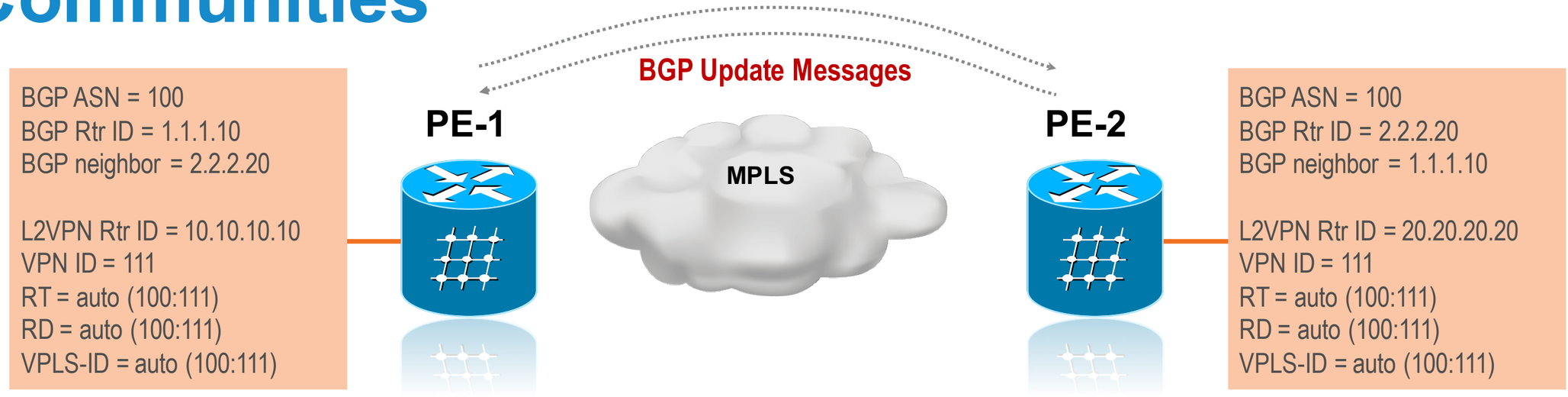
BGP Auto-Discovery (BGP-AD)

- Eliminates need to manually provision VPLS neighbors
- Automatically detects when new PEs are added / removed from the VPLS domain
- Uses BGP Update messages to advertize PE/VFI mapping (VPLS NLRI)
- Typically used in conjunction with BGP Route Reflectors to minimize iBGP full-mesh peering requirements
- Two (2) RFCs define use of BGP for VPLS AD[1]
  - RFC 6074 – when LDP used for PW signaling
  - RFC 4761 – when BGP used for PW signaling



(1) VPLS BGP NLRIs from RFC 6074 and 4761 are different in format and thus not compatible, even though they share same AFI / SAFI values

What is Discovered? NLRI + Extended Communities



	Source Address = 1.1.1.10	Source Address = 2.2.2.20
	Destination Address = 2.2.2.20	Destination Address = 1.1.1.10
NLRI	Length = 14	Length = 14
NLRI	Route Distinguisher = 100:111	Route Distinguisher = 100:111
NLRI	L2VPN Router ID = 10.10.10.10	L2VPN Router ID = 20.20.20.20
Extended Communities	VPLS-ID = 100:111	VPLS-ID = 100:111
Extended Communities	Route Target = 100:111	Route Target = 100:111

VPLS LDP Signaling and BGP-AD

Cisco IOS

BGP Auto-Discovery attributes
VPLS VFI attributes
Signaling attributes

```
hostname PE1
!
interface Loopback0
 ip address 102.102.102.102 255.255.255.255
!
router bgp 100
 bgp router-id 102.102.102.102
 neighbor 104.104.104.104 remote-as 100
 neighbor 104.104.104.104 update-source Loopback0
 !
 address-family l2vpn vpls
  neighbor 104.104.104.104 activate
  neighbor 104.104.104.104 send-community extended
 exit-address-family
```

BGP L2VPN AF

```
l2 vfi sample-vfi autodiscovery
 vpn id 300
 vpls-id 100:300
!
interface Vlan300
 xconnect vfi sample-vfi
```

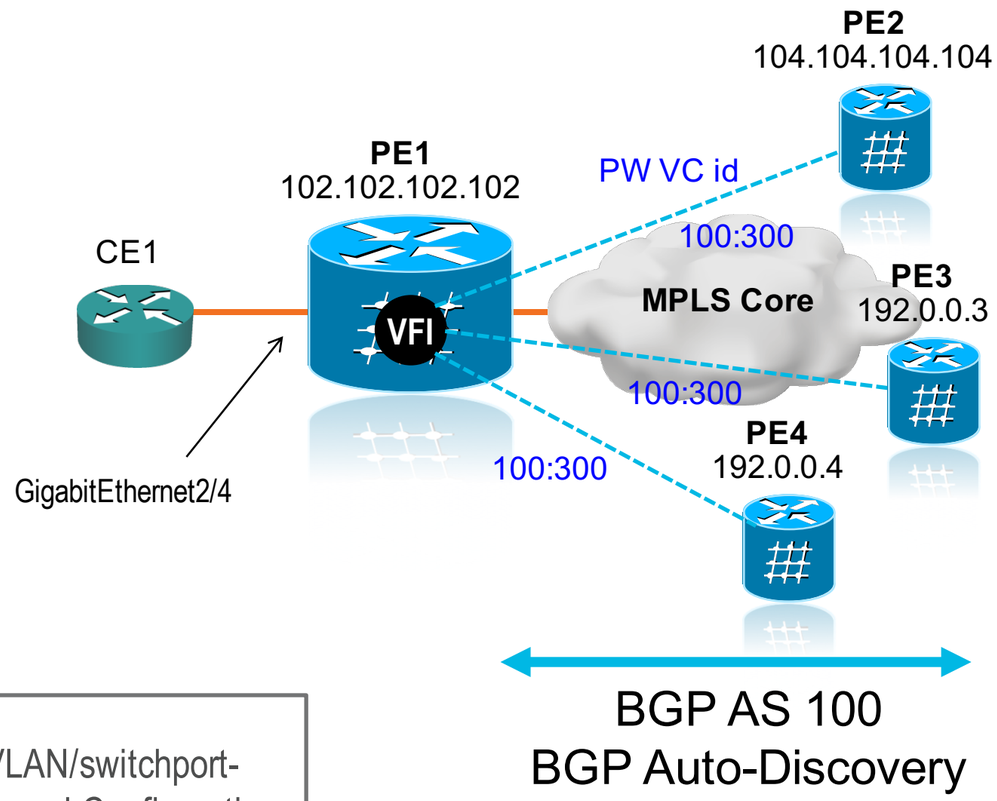
Bridge Domain-based Configuration

OR

VLAN/switchport-based Configuration

```
interface GigabitEthernet2/4
 service instance 333 ethernet
  encapsulation dot1q 333
  rewrite ingress tag pop 1 symmetric
  bridge-domain 300
```

```
interface GigabitEthernet2/4
 switchport mode trunk
 switchport trunk allowed vlan 300
```



VPLS LDP Signaling and BGP-AD

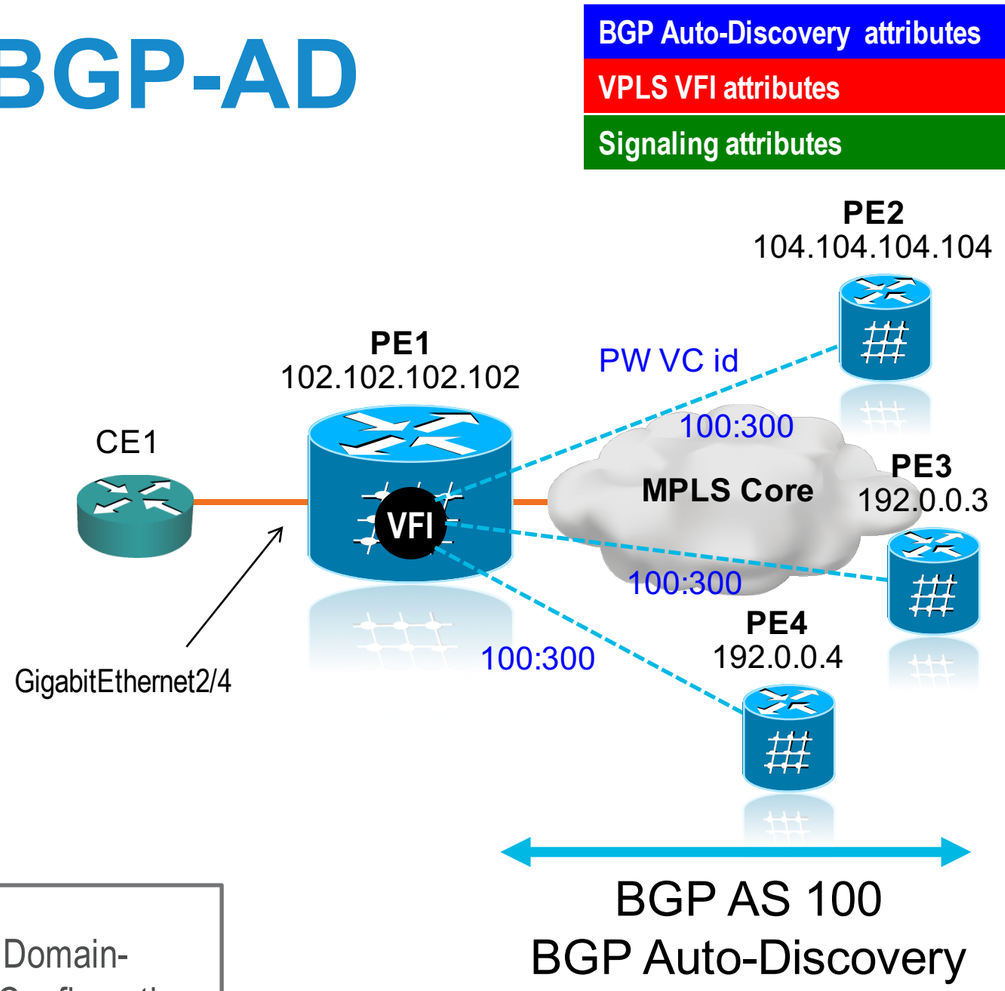
Cisco IOS (NEW Protocol-based CLI)

```
hostname PE1
!
interface Loopback0
 ip address 102.102.102.102 255.255.255.255
!
router bgp 100
 bgp router-id 102.102.102.102
 neighbor 104.104.104.104 remote-as 100
 neighbor 104.104.104.104 update-source Loopback0
 !
 address-family l2vpn vpls
  neighbor 104.104.104.104 activate
  neighbor 104.104.104.104 send-community extended
 exit-address-family
```

```
l2vpn vfi context sample-vfi
  vpn id 300
  autodiscovery bgp signaling ldp
   vpls-id 100:300
!
bridge-domain 300
 member vfi sample-vfi
 member GigabitEthernet2/4 service instance 333
```

Bridge Domain-based Configuration

```
interface GigabitEthernet2/4
 service instance 333 ethernet
  encapsulation dot1q 333
  rewrite ingress tag pop 1 symmetric
```



VPLS LDP Signaling and BGP-AD

Cisco IOS XR

BGP Auto-Discovery attributes
VPLS VFI attributes
Signaling attributes

```
hostname PE1
!
interface Loopback0
 ipv4 address 106.106.106.106 255.255.255.255
!
interface GigabitEthernet0/0/0/2.101 l2transport
 encapsulation dot1q 101
 rewrite ingress tag pop 1 symmetric
```

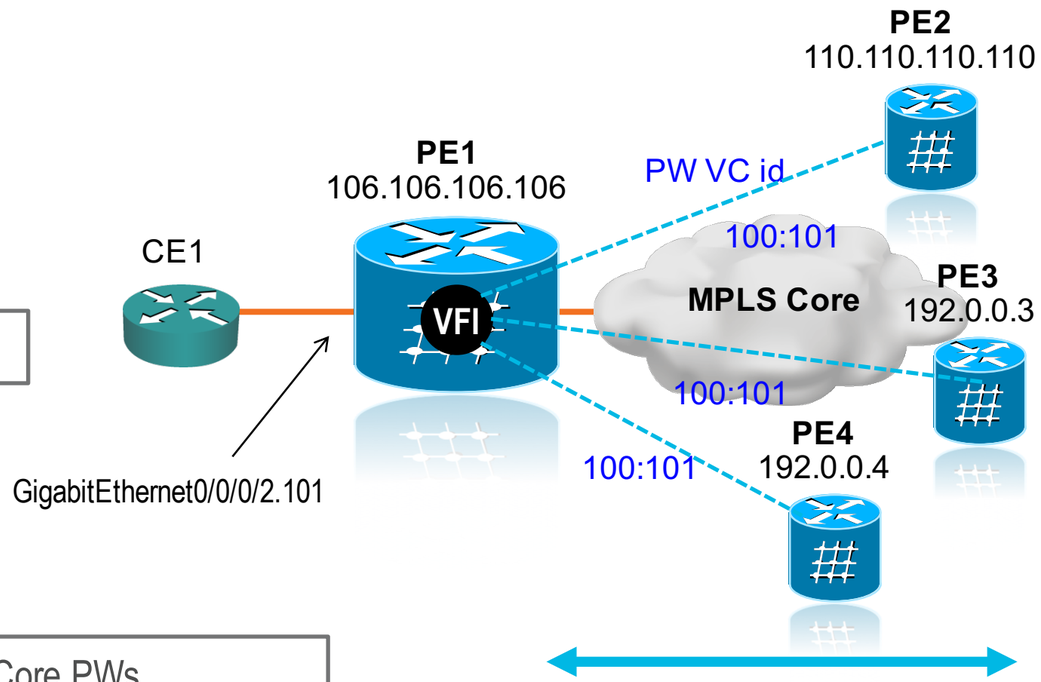
```
router bgp 100
 bgp router-id 106.106.106.106
 address-family l2vpn vpls-vpws
 neighbor 110.110.110.110
  remote-as 100
  update-source Loopback0
  address-family l2vpn vpls-vpws
```

BGP L2VPN AF

```
l2vpn
 bridge group Cisco-Live
  bridge-domain bd101
   interface GigabitEthernet0/0/0/2.101
   vfi vfi101
    vpn-id 11101
    autodiscovery bgp
     rd auto
     route-target 100:101
     signaling-protocol ldp
      vpls-id 100:101
```

Full-mesh Core PWs auto-discovered with BGP-AD and signaled by LDP

PW ID = VPLS-id (100:101)



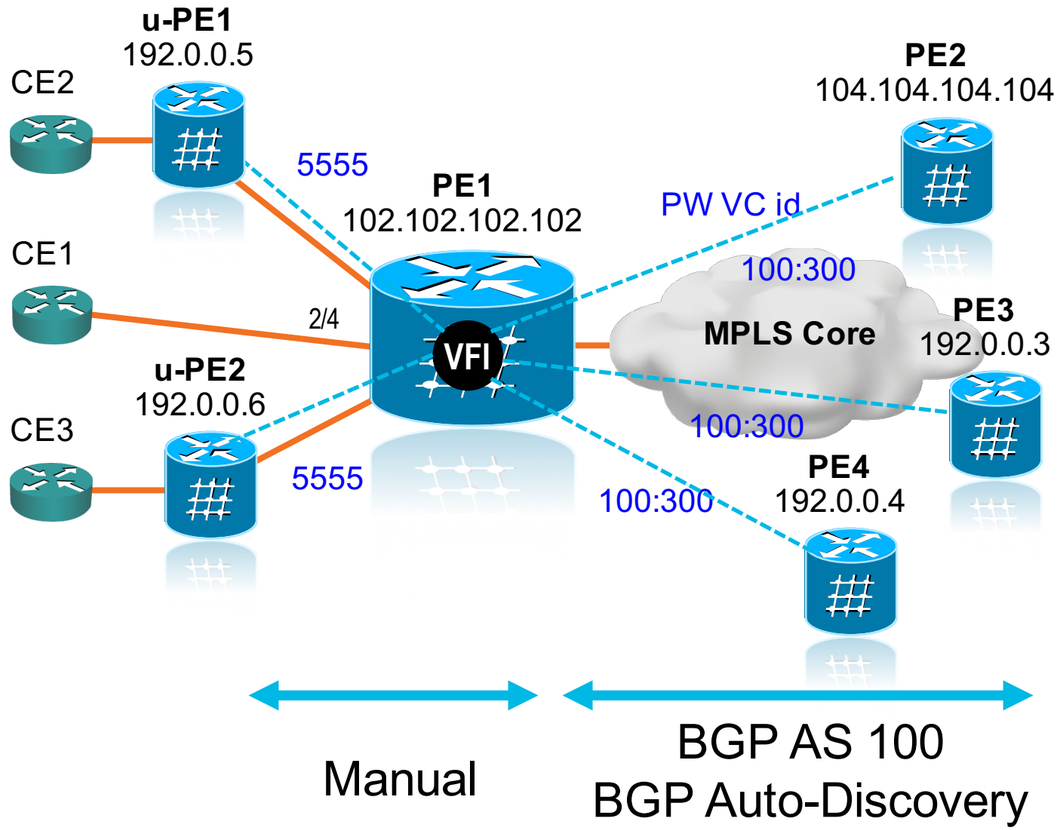
BGP AS 100
BGP Auto-Discovery

H-VPLS LDP Signaling and BGP-AD / Manual provisioning

Cisco IOS

```
hostname PE1
!
interface Loopback0
 ip address 102.102.102.102 255.255.255.255
!
l2 vfi sample-vfi autodiscovery
 vpn id 300
 vpls-id 100:300
 neighbor 192.0.0.5 5555 encapsulation mpls no-split-horizon
 neighbor 192.0.0.6 5555 encapsulation mpls no-split-horizon
```

Manually provisioned Spoke PWs

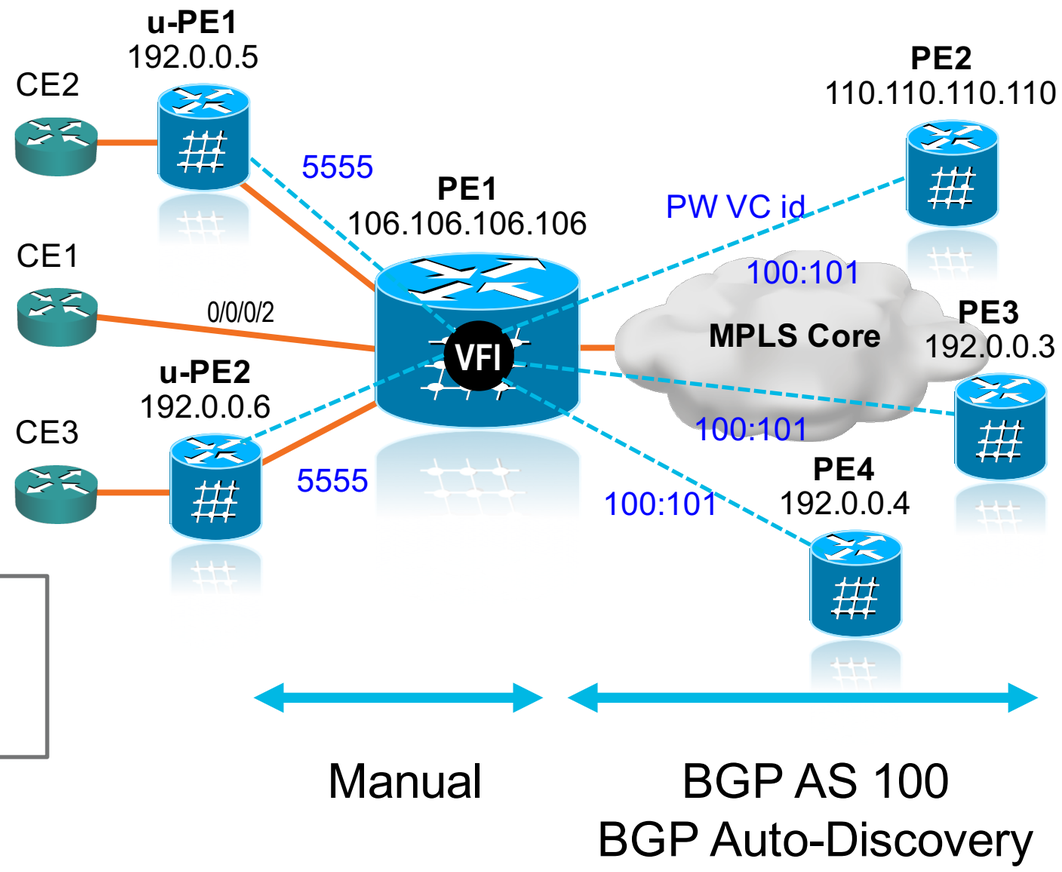


H-VPLS LDP Signaling and BGP-AD / Manual provisioning

Cisco IOS XR

```
hostname PE1
!
l2vpn
 bridge group Cisco-Live
  bridge-domain bd101
   interface GigabitEthernet0/0/0/2.101
   !
   neighbor 192.0.0.5 pw-id 5555
   !
   neighbor 192.0.0.6 pw-id 5555
   !
   vfi vfi101
    vpn-id 11101
    autodiscovery bgp
     rd auto
     route-target 100:101
     signaling-protocol ldp
      vpls-id 100:101
```

Manually provisioned Spoke PWs



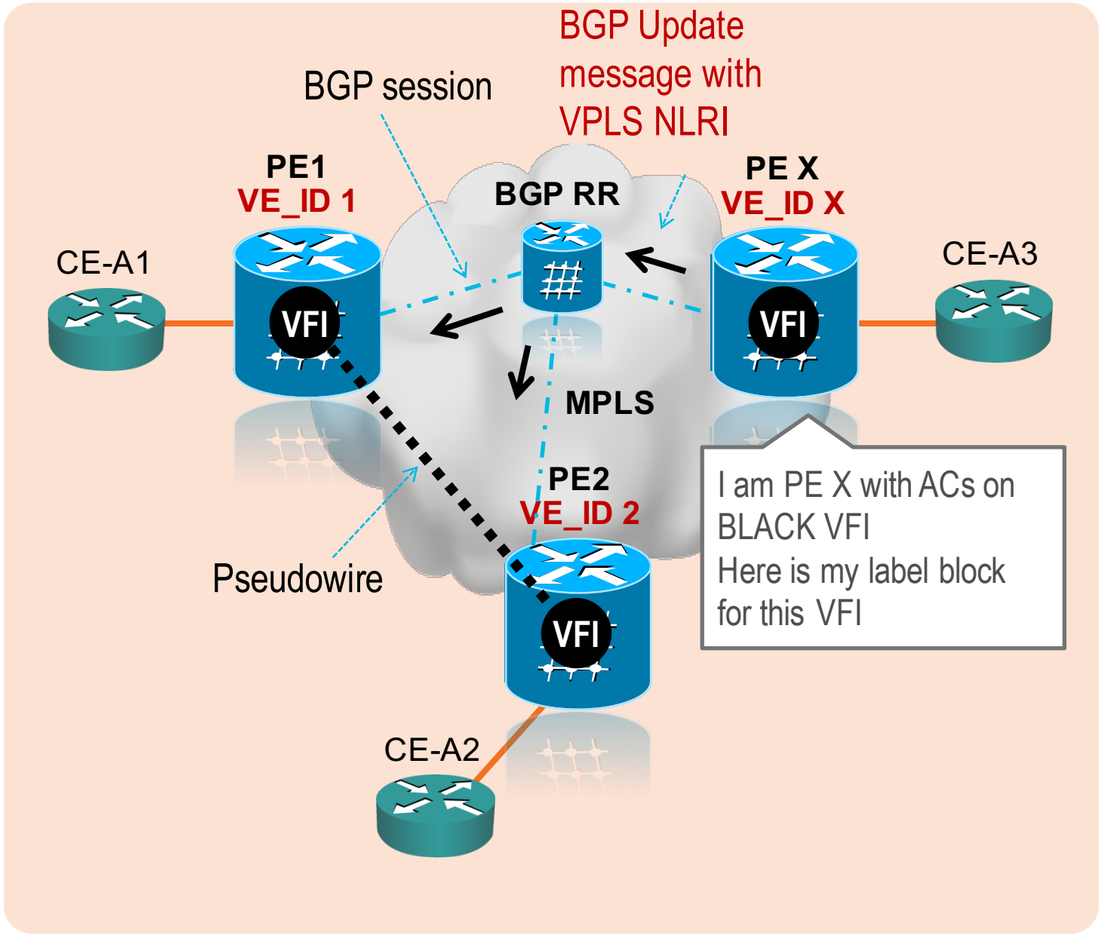
Pseudowire (PW) Signaling and PE Auto-Discovery

BGP-based AutoDiscovery (BGP-AD) and BGP Signaling

BGP Signaling and Auto-Discovery

Overview

- RFC 4761[1] defines use of BGP for VPLS PE Auto-Discovery and Signaling
- All PEs within a given VPLS are assigned a unique VPLS Edge device ID (VE ID)
- A PE X wishing to send a VPLS update sends the same label block information to all other PEs using BGP VPLS NLRI
- Each receiving PE infers the label intended for PE X by adding its (unique) VE ID to the label base
  - Each receiving PE gets a unique label for PE X for that VPLS



(1) VPLS BGP NLRIs from RFC 6074 and 4761 are different in format and thus not compatible, even though they share same AFI / SAFI values

VPLS BGP Signaling and BGP-AD

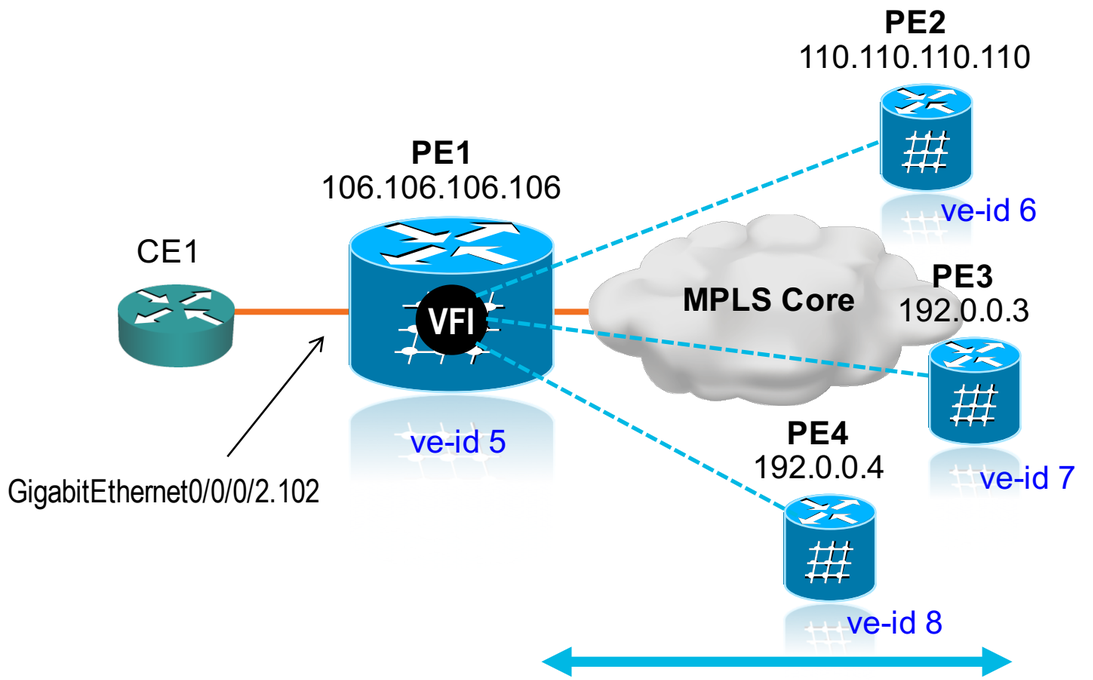
Cisco IOS XR

BGP Auto-Discovery attributes
VPLS VFI attributes
Signaling attributes

```
hostname PE1
!
interface Loopback0
 ipv4 address 106.106.106.106 255.255.255.255
!
router bgp 100
 bgp router-id 106.106.106.106
 address-family l2vpn vpls-vpws
 neighbor 110.110.110.110
  remote-as 100
  update-source Loopback0
  address-family l2vpn vpls-vpws
```

```
l2vpn
 bridge group Cisco-Live
  bridge-domain bd102
   interface GigabitEthernet0/0/0/2.102
   vfi vfi102
    vpn-id 11102
    autodiscovery bgp
     rd auto
     route-target 100:102
     signaling-protocol bgp
      ve-id 5
```

VE-id must be unique in a VPLS instance



BGP AS 100
BGP Signaling and Auto-Discovery

VPLS BGP Signaling and BGP-AD

Cisco IOS (NEW Protocol-based CLI)

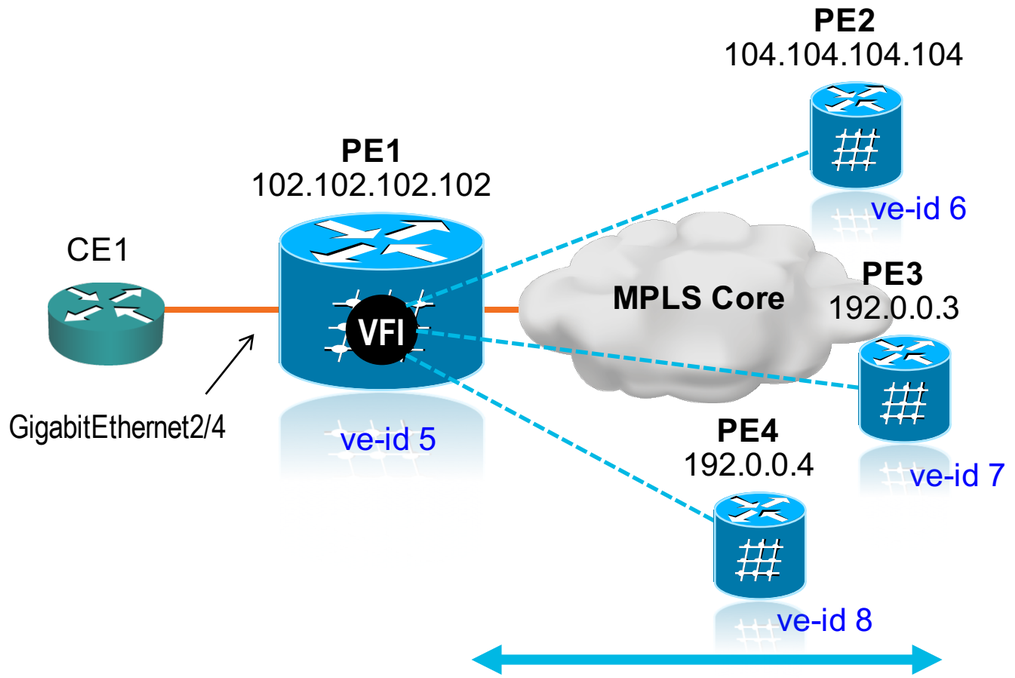
```
hostname PE1
!
interface Loopback0
 ip address 102.102.102.102 255.255.255.255
!
router bgp 100
 bgp router-id 102.102.102.102
 neighbor 104.104.104.104 remote-as 100
 neighbor 104.104.104.104 update-source Loopback0
 !
 address-family l2vpn vpls
  neighbor 104.104.104.104 activate
  neighbor 104.104.104.104 send-community extended
  neighbor 104.104.104.104 suppress-signaling-protocol ldp
 exit-address-family
```

```
l2vpn vfi context sample-vfi
  vpn id 3300
  autodiscovery bgp signaling bgp
    ve id 5
    ve range 10
```

VE-id must be unique in a VPLS instance

```
bridge-domain 300
 member vfi sample-vfi
 member GigabitEthernet2/4 service instance 333
!
interface GigabitEthernet2/4
 service instance 333 ethernet
  encapsulation dot1q 300
  rewrite ingress tag pop 1 symmetric
```

Bridge Domain-based Configuration



BGP AS 100

BGP Signaling and Auto-Discovery

BUILT FOR
THE HUMAN
NETWORK

Cisco live!