MPLS Workshop
Day 4

Jan 15th to Jan 19th 2009

Abdul Rahim , arahim@cisco.com

Configuring RIP as a Routing Protocol Between PE and CE Routers

Outline

Configuring RIP PE-CE Routing

Avoiding Routing Loops with RIP as PE-CE Protocol

Configuring RIP PE-CE Routing

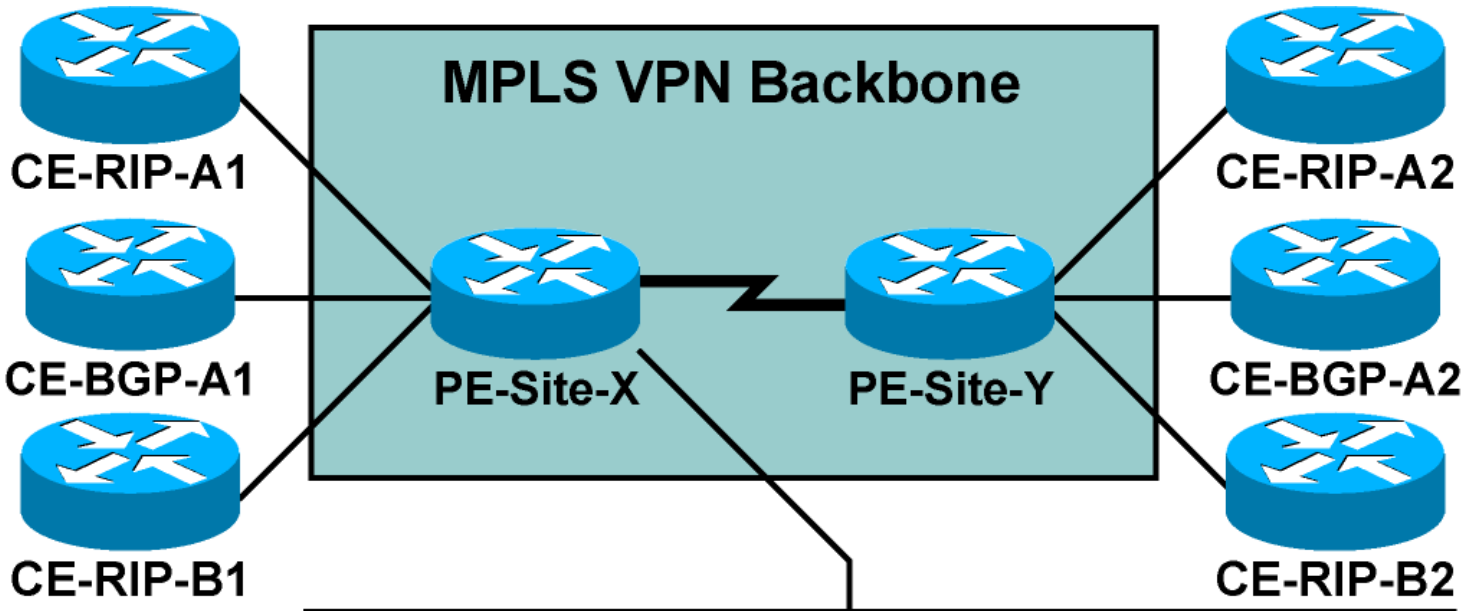
- A routing context is configured for each VRF running RIP.
- RIP parameters have to be specified in the VRF.
- Some parameters configured in the RIP process are propagated to routing contexts
  (for example, RIP version).
- Only RIPv2 is supported.
- RIP may work but does not support VLSM (Variable Length Subnet Mask)

Configuring RIP PE-CE Routing (Cont.)
RIP Metric Propagation

```
router rip
 version 2
  address-family ipv4 vrf vrf-name
    version 2
    redistribute bgp as-number metric transparent
```

- BGP routes must be redistributed back into RIP.
- The RIP hop count has to be manually set for routes redistributed into RIP.
- For end-to-end RIP networks, the following applies:
  - On the sending end, the RIP hop count is copied into the BGP multi-exit discriminator attribute (default BGP behavior).
  - On the receiving end, the **metric transparent** option copies the BGP MED into the RIP hop count, resulting in a consistent end-to-end RIP hop count. This hop count does not have the hops traversed via the MPLS VPN backbone
- When you are using RIP with other protocols, the metric must be manually set.

Configuring RIP PE-CE Routing (Cont.)




```
router rip
 version 2
 address-family ipv4 vrf Customer_ABC
  network 10.0.0.0
  redistribute bgp 12703 metric transparent
!
router bgp 12703
 address-family ipv4 vrf Customer_ABC
  redistribute rip
  no auto-summary
```

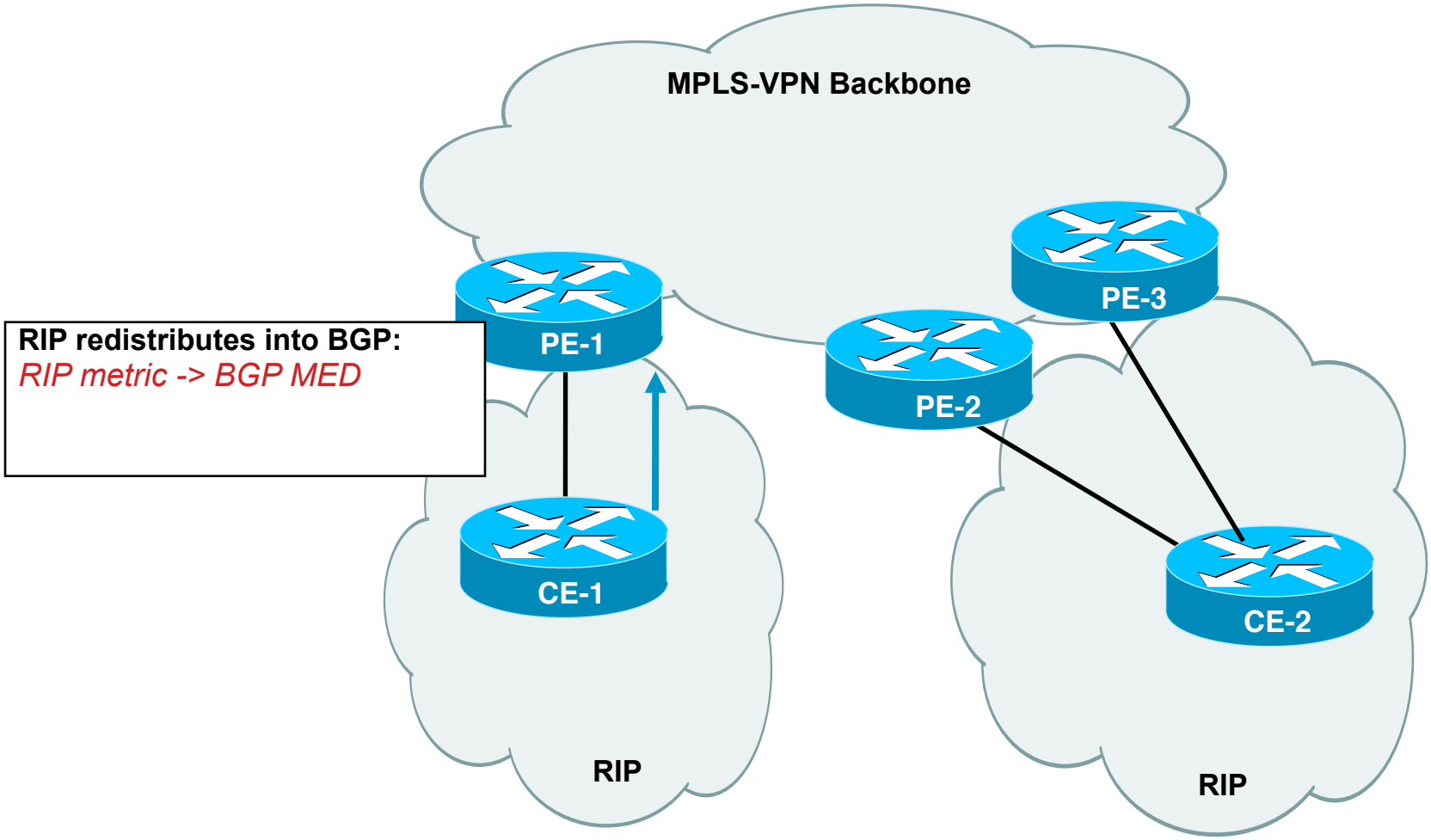
020G_073

Loop Detection with RIP as PE-CE

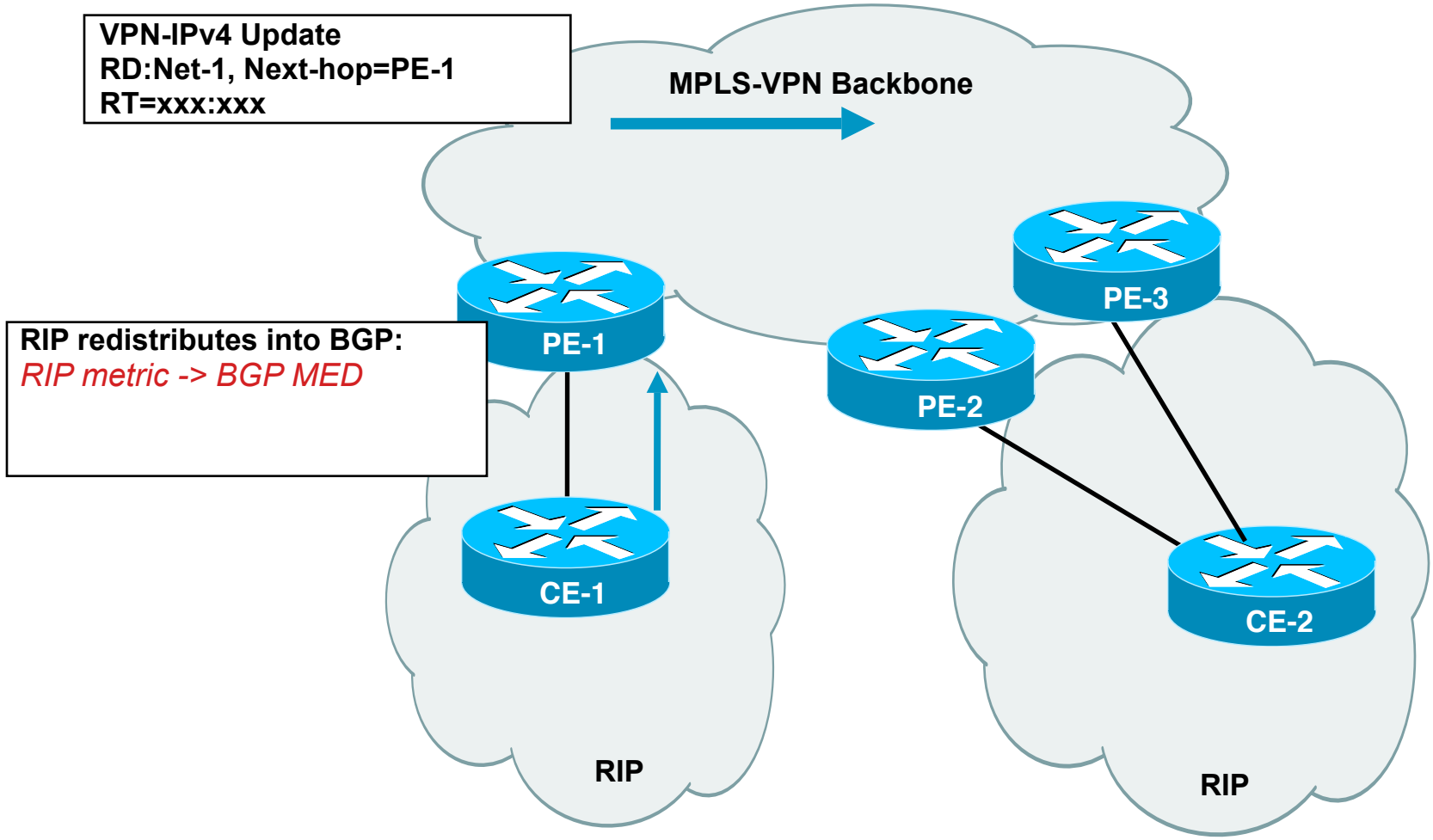
RIP works with the following mechanisms for loop detection:

- Split Horizon
- Site Of Origin (SOO)

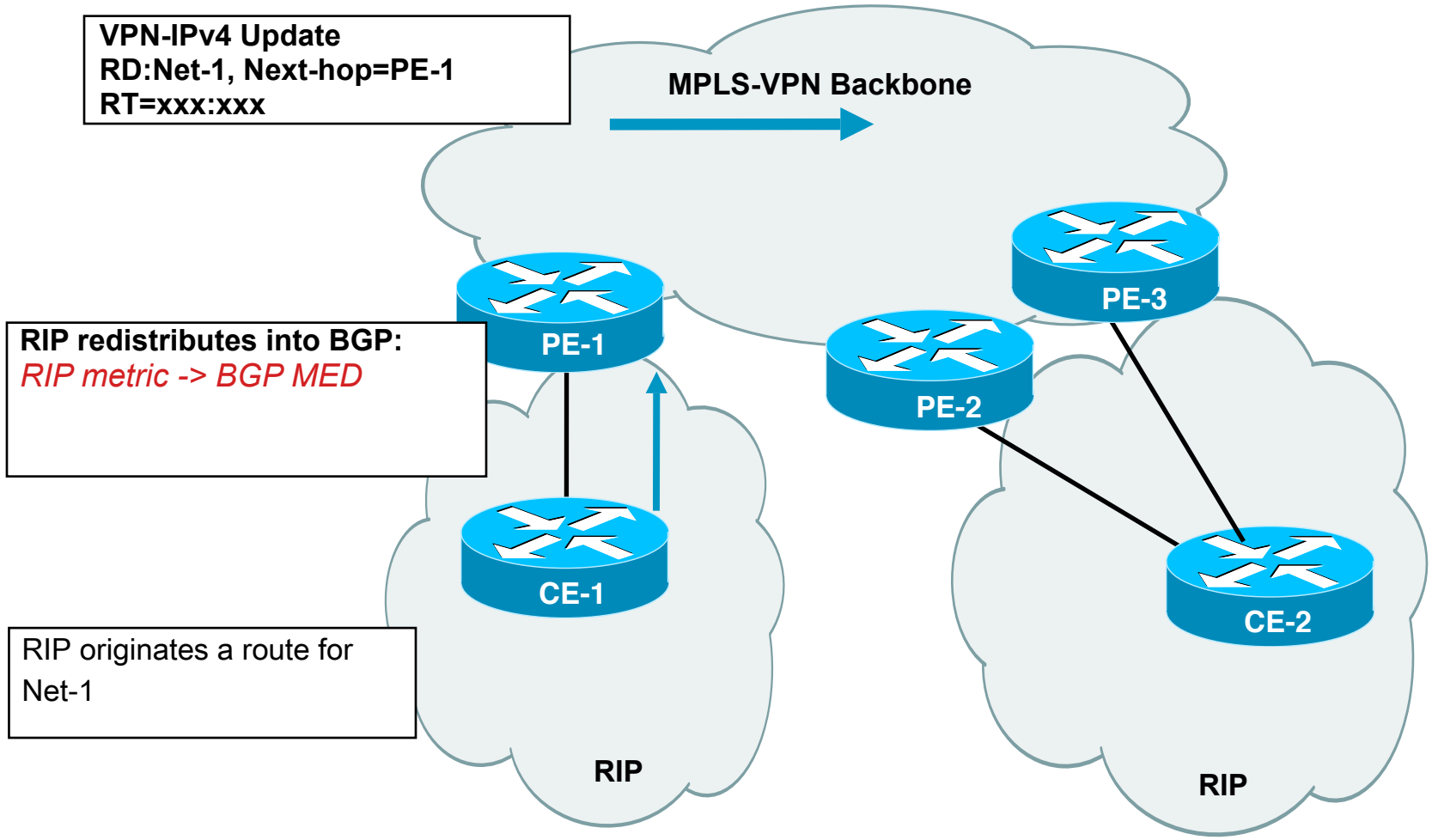
Avoiding Routing Loops: Split-horizon



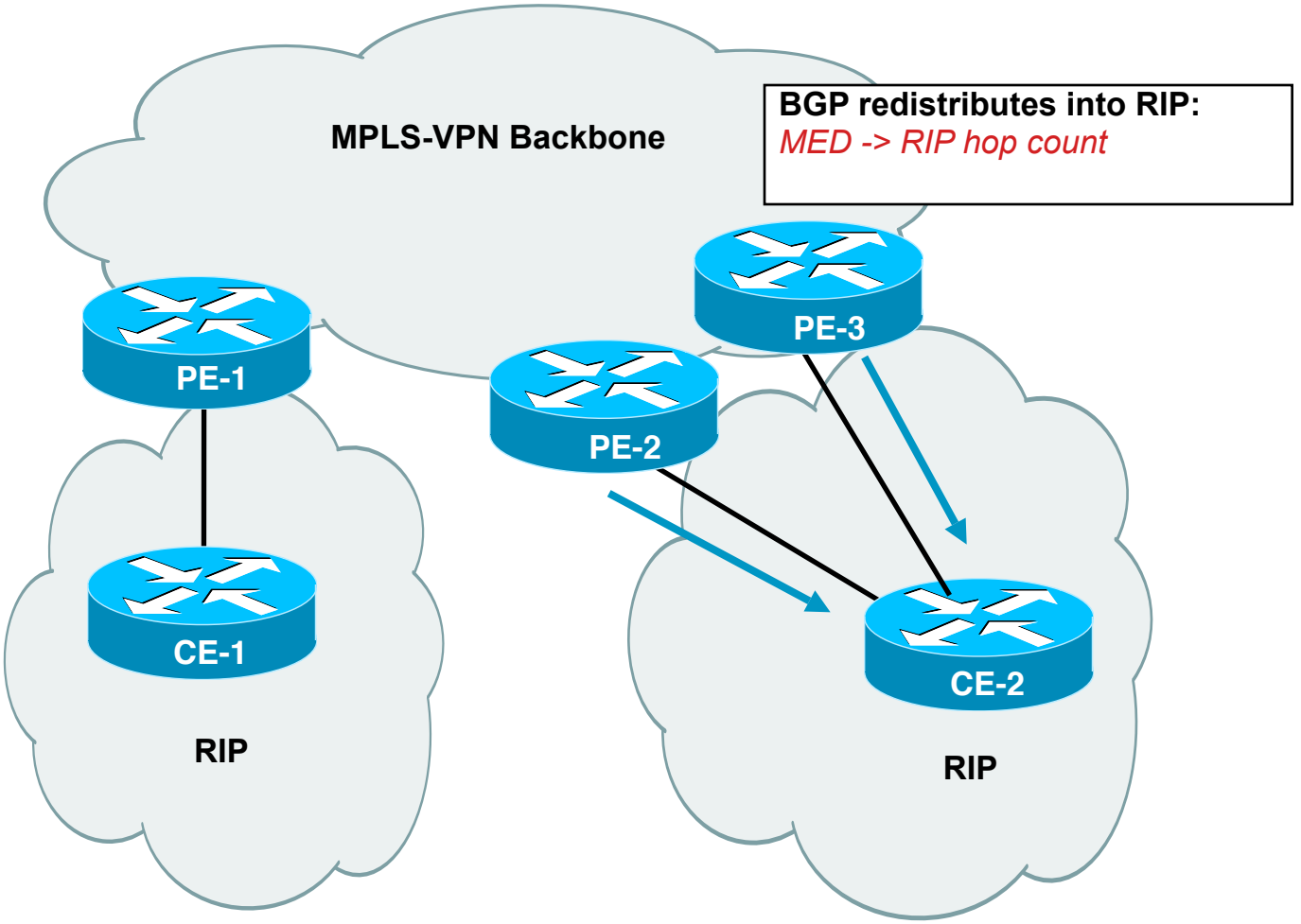
Avoiding Routing Loops: Split-horizon



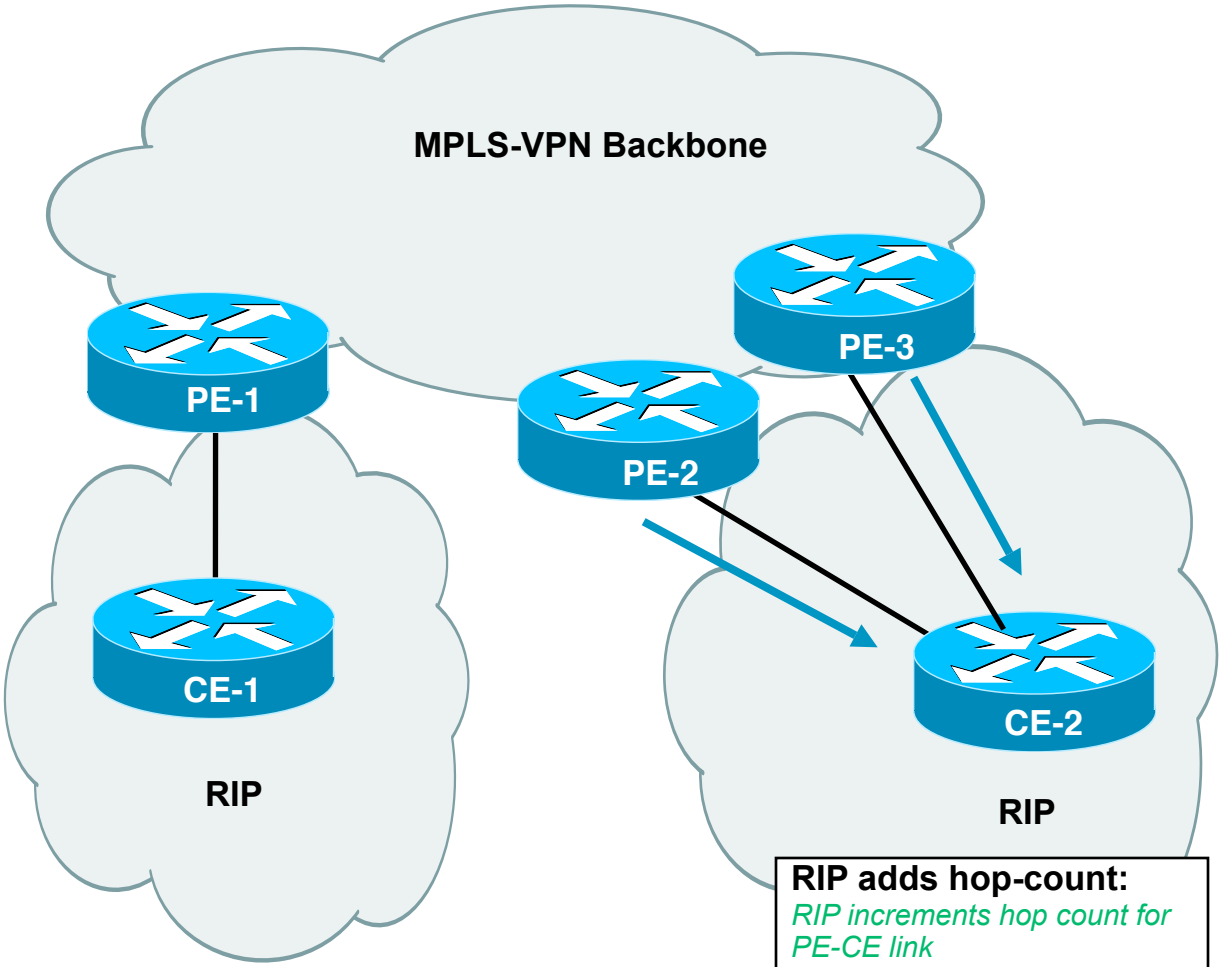
Avoiding Routing Loops: Split-horizon



Avoiding Routing Loops: Split-horizon

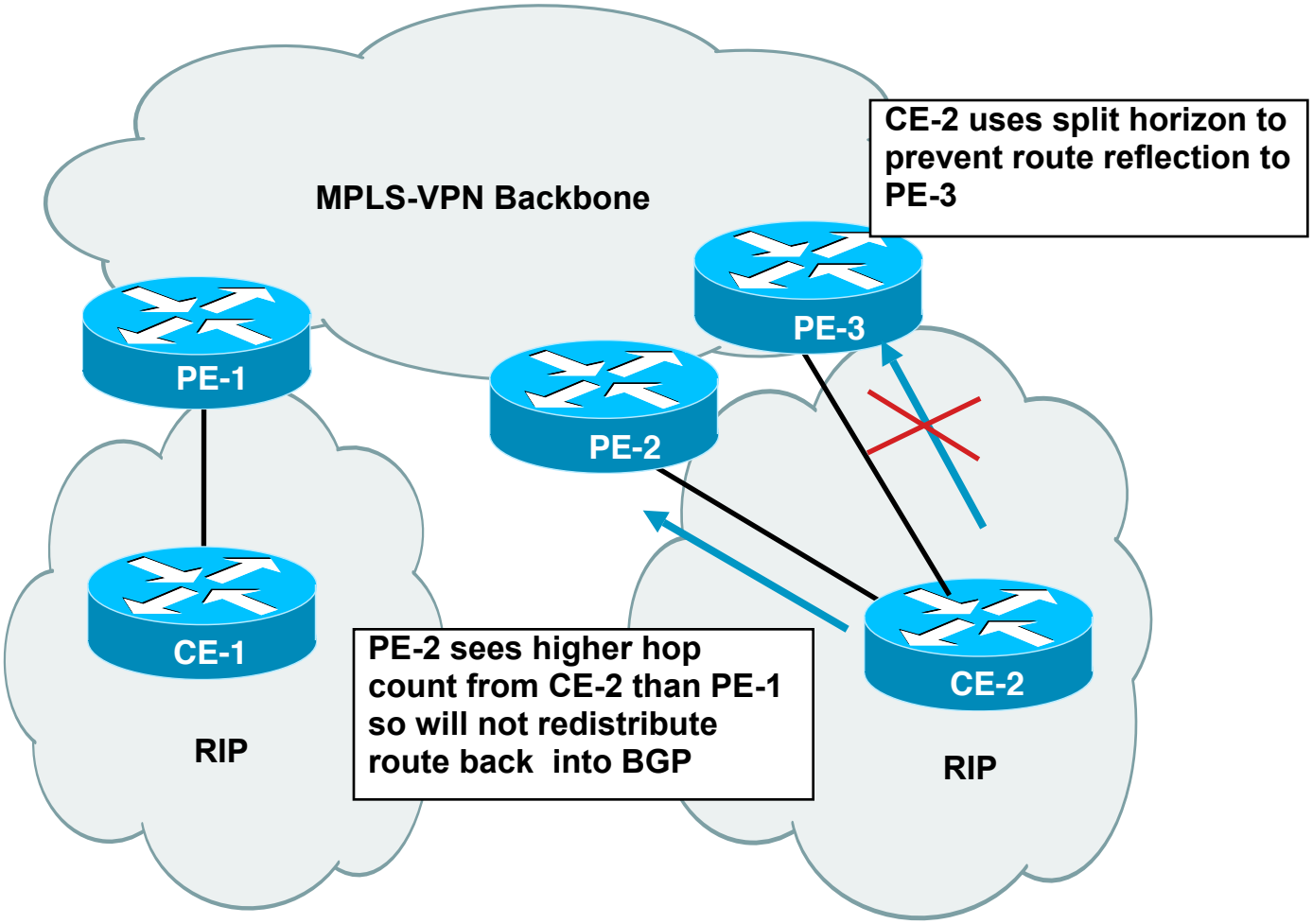


Avoiding Routing Loops: Split-horizon

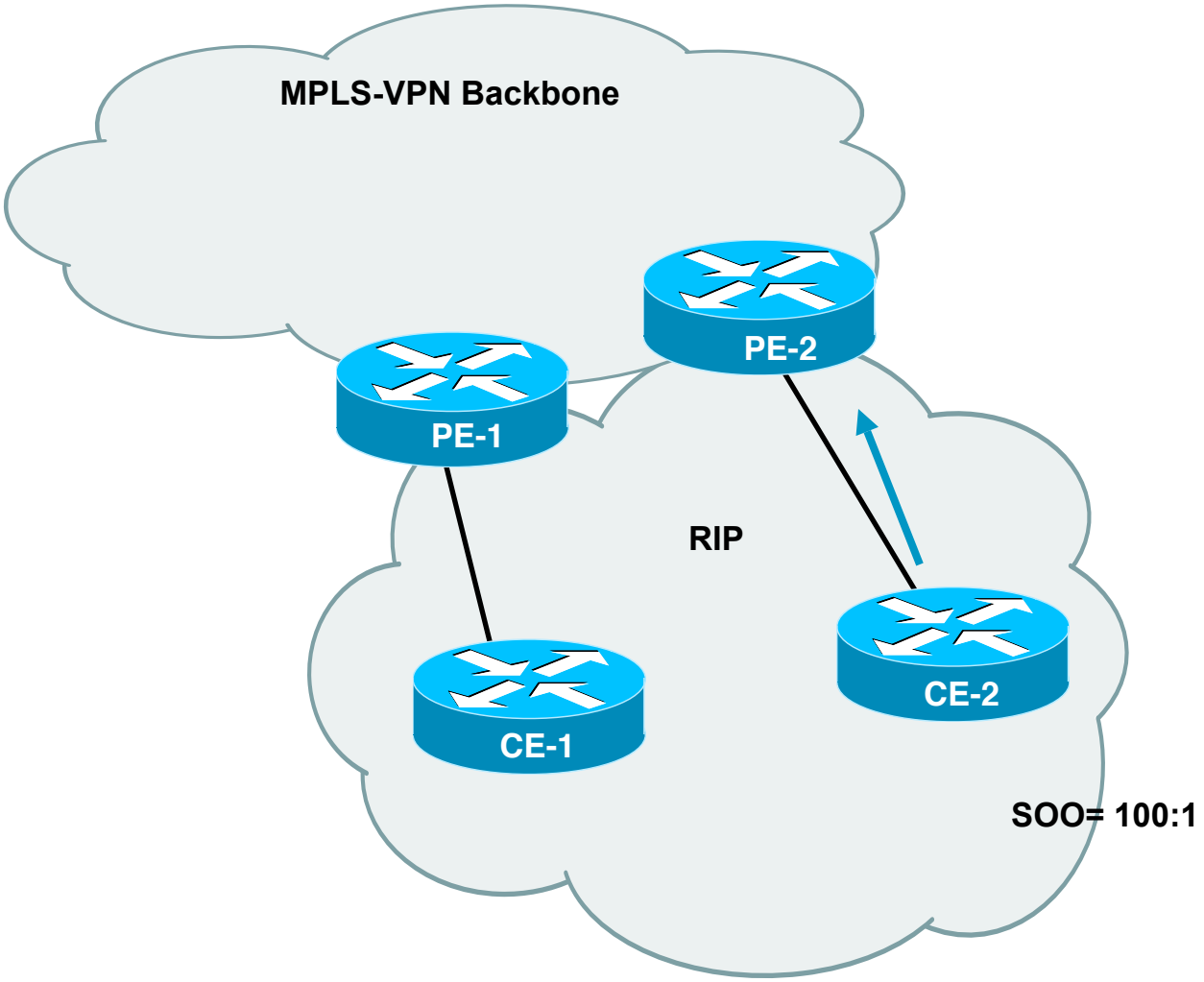


RIP adds hop-count:
RIP increments hop count for PE-CE link

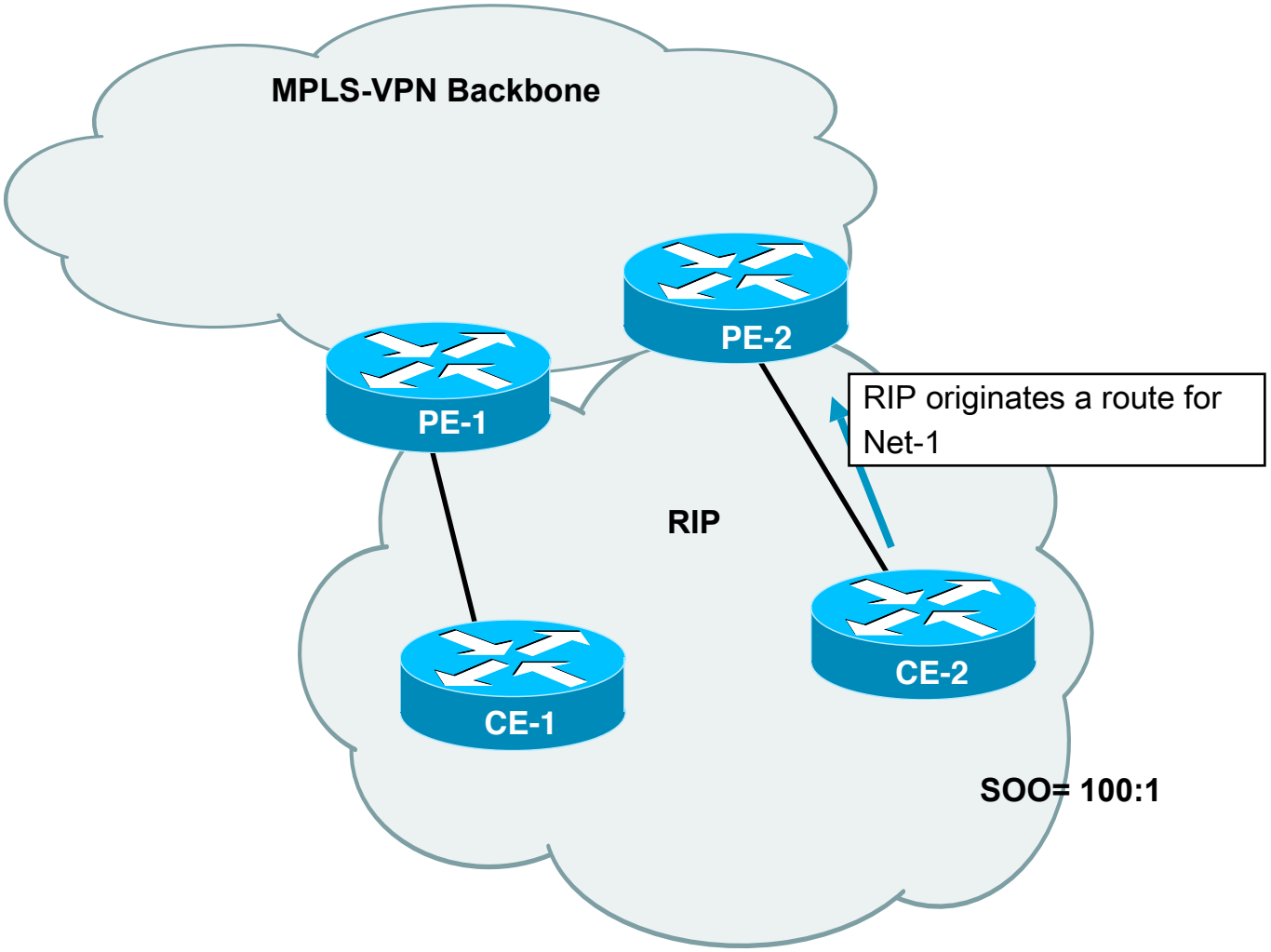
Avoiding Routing Loops: Split-horizon



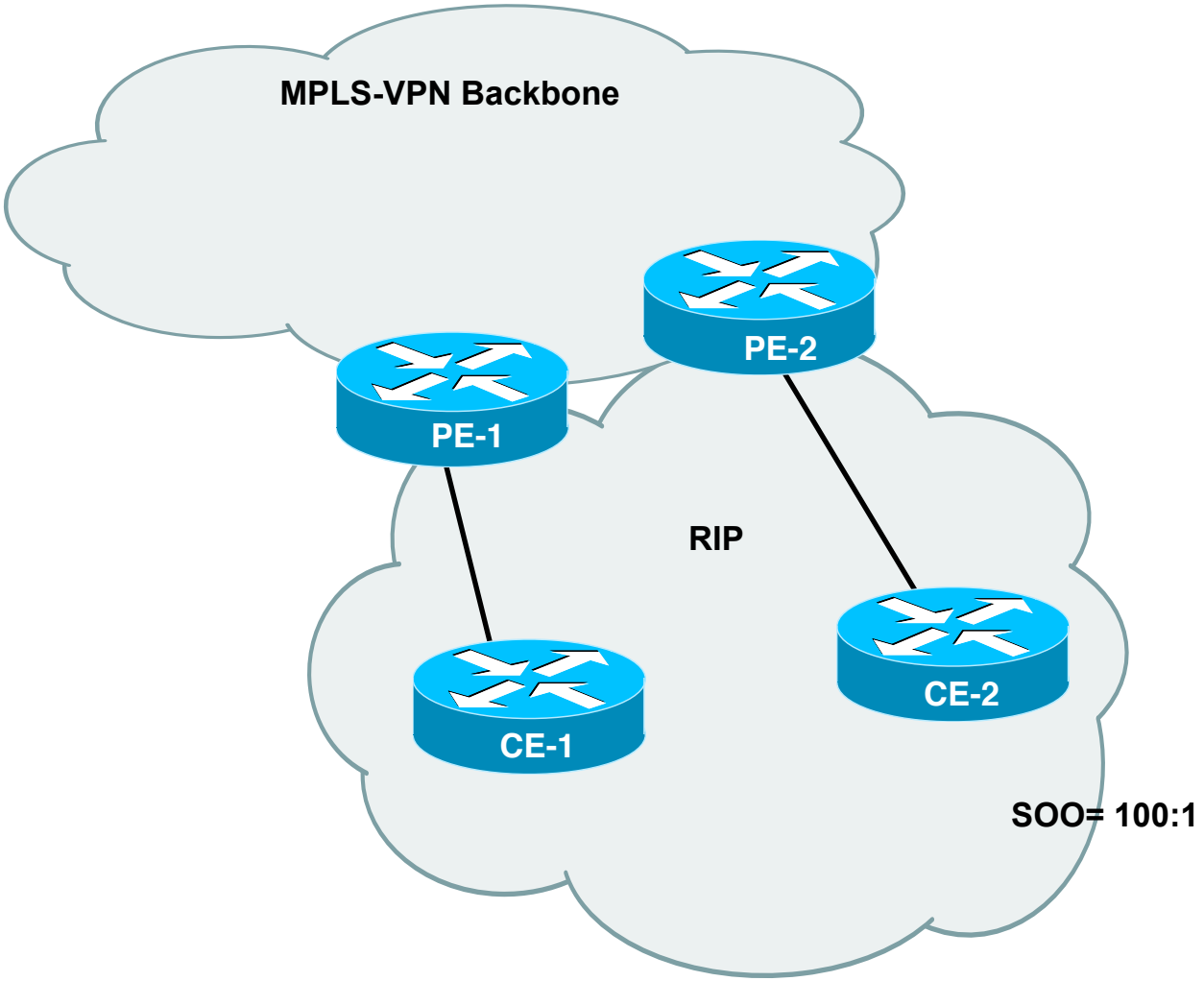
Avoiding Routing Loops: Site Of Origin



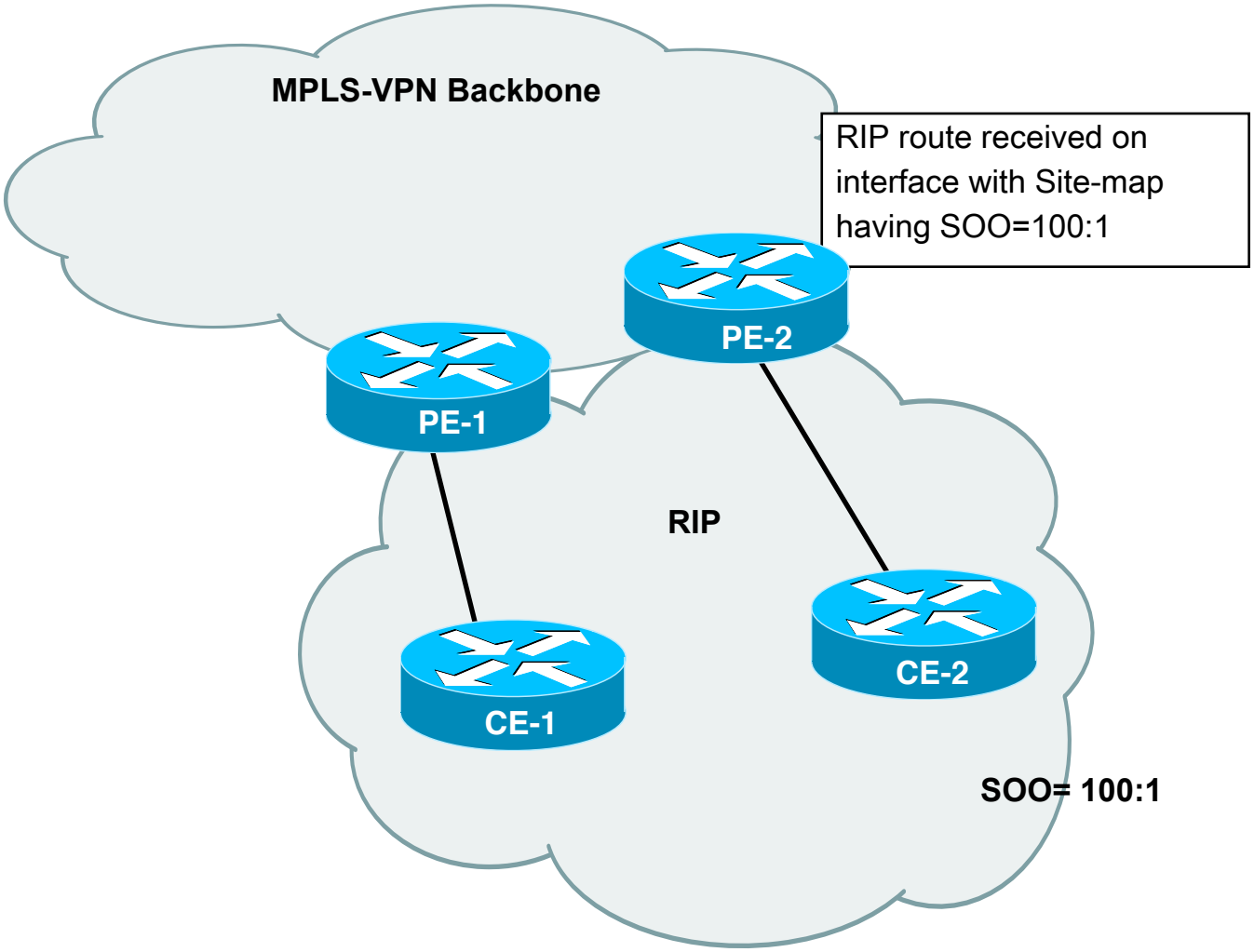
Avoiding Routing Loops: Site Of Origin



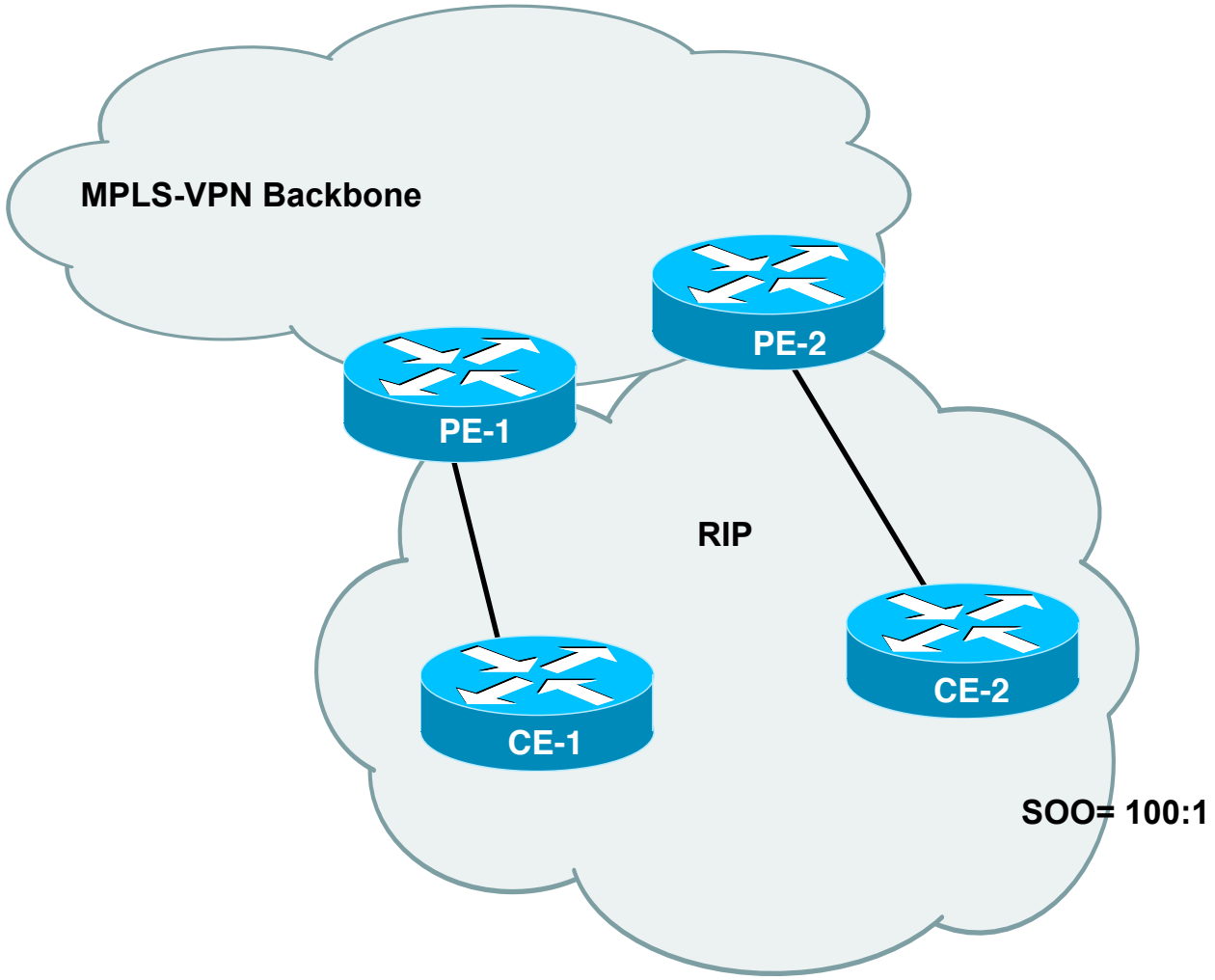
Avoiding Routing Loops: Site Of Origin



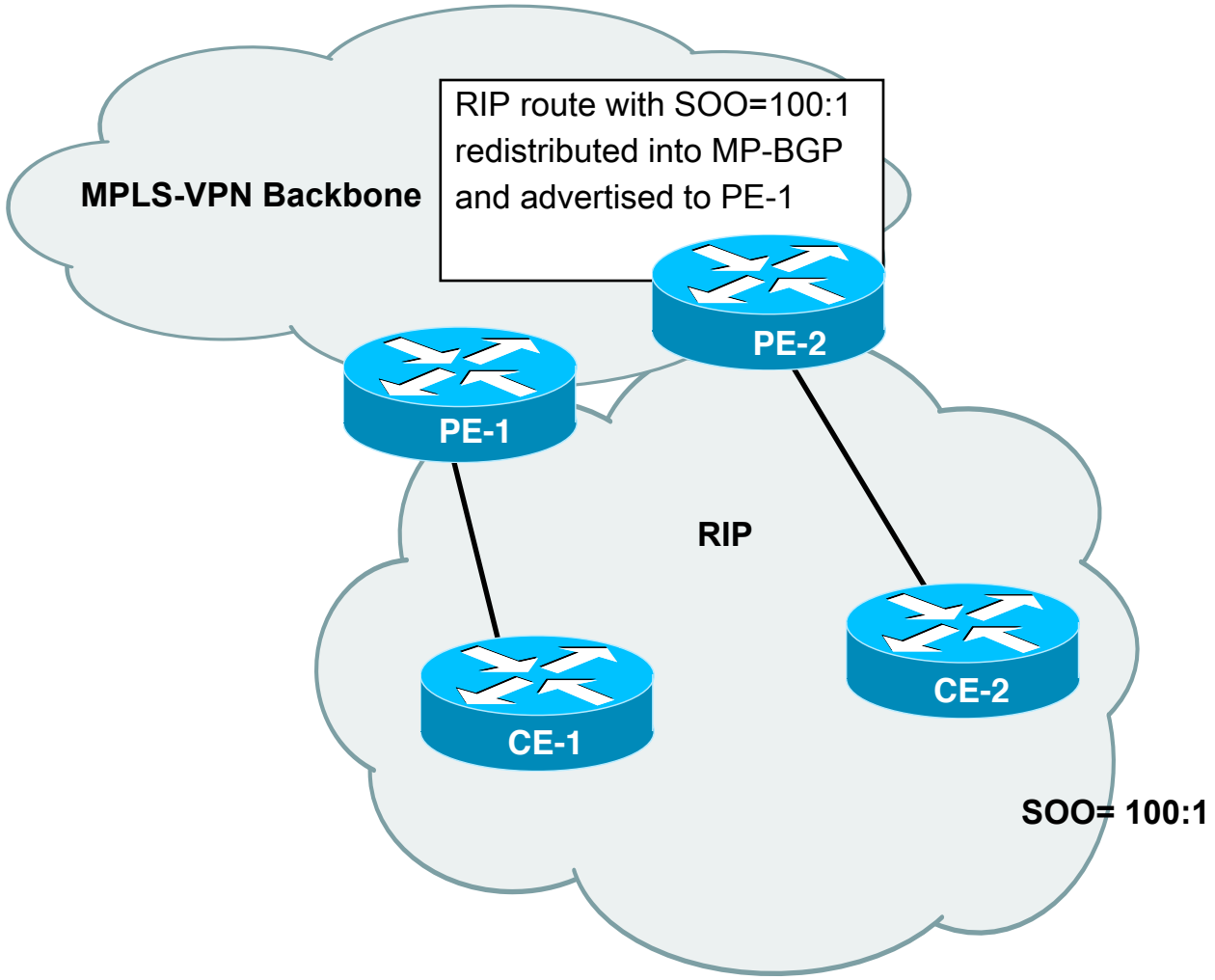
Avoiding Routing Loops: Site Of Origin



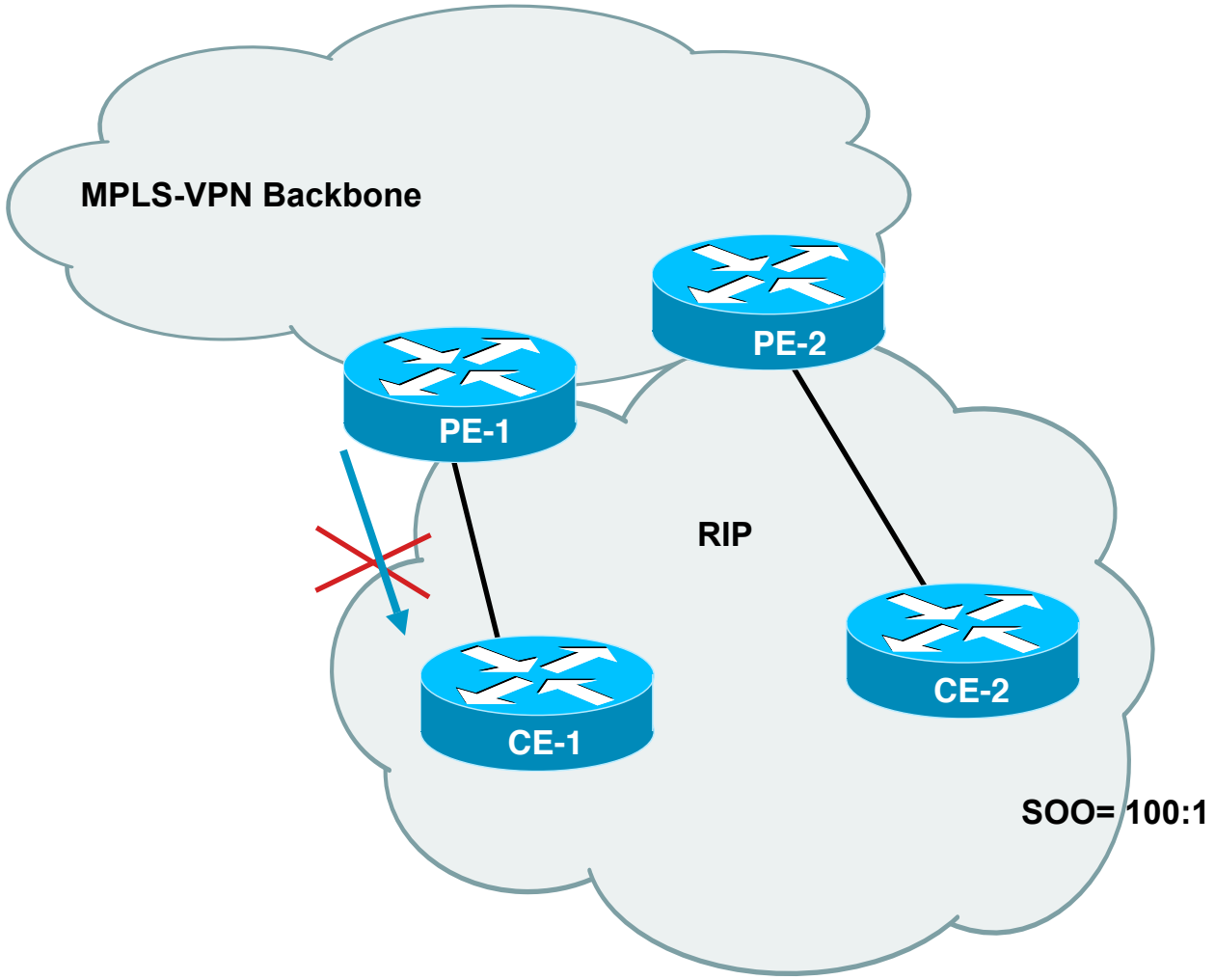
Avoiding Routing Loops: Site Of Origin



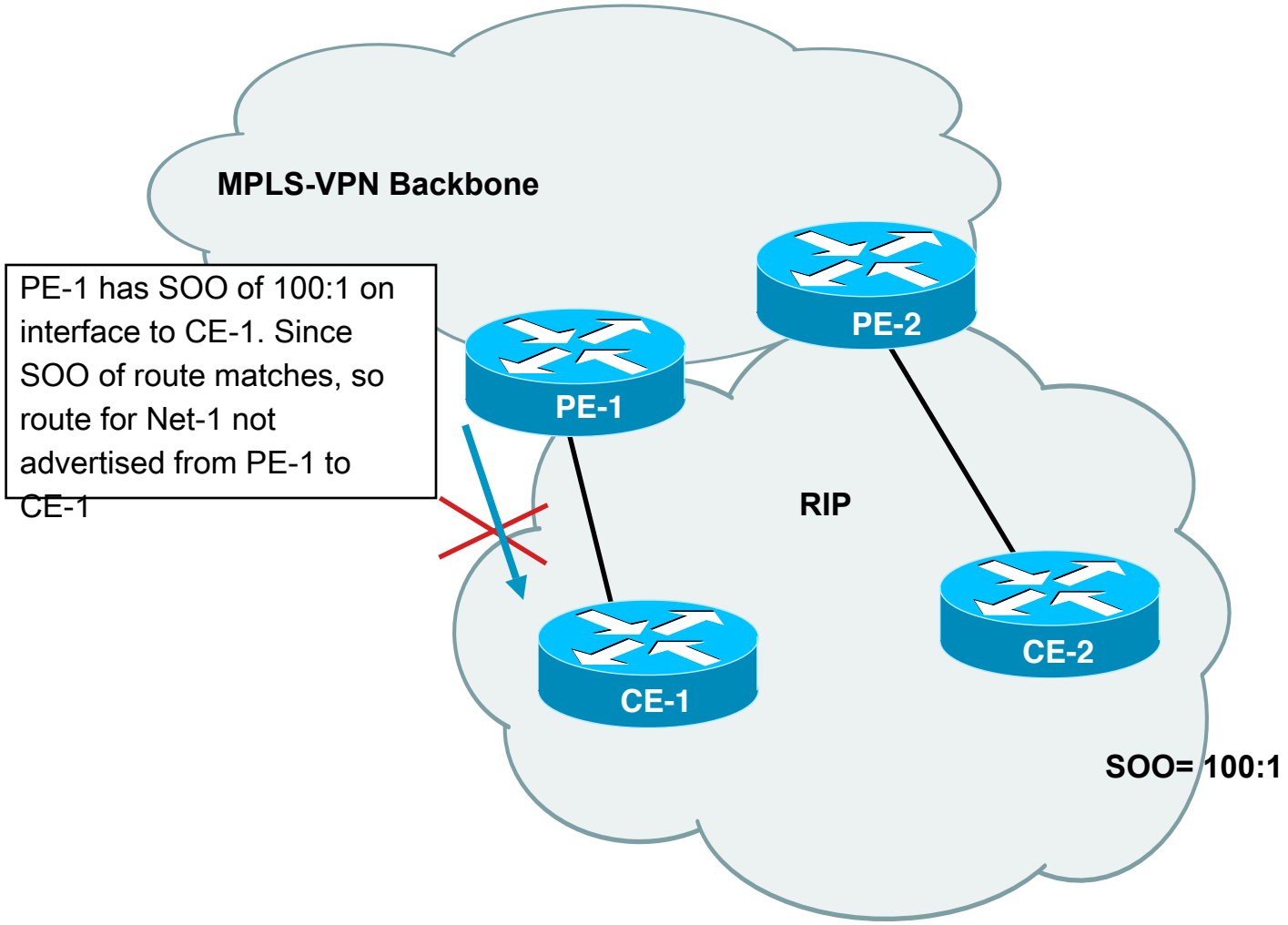
Avoiding Routing Loops: Site Of Origin



Avoiding Routing Loops: Site Of Origin



Avoiding Routing Loops: Site Of Origin



Summary

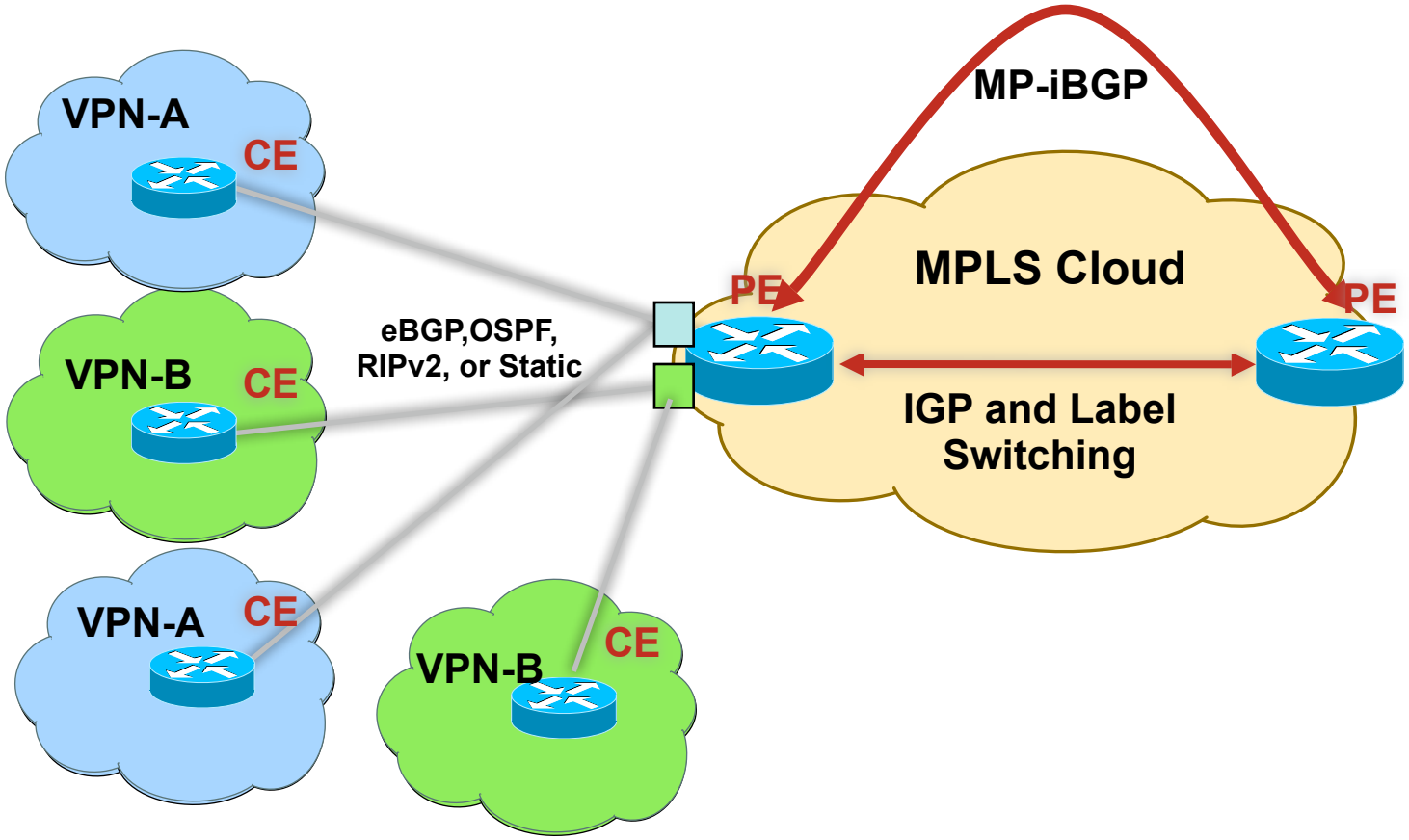
- RIP can be used as a PE-CE routing protocol
- RIP v2 should be used as it supports VLSM
- RIP has loop detection mechanisms to prevent routing loops with complex connectivity models

Multi-VRF CE (aka VRF-lite)

What is Multi-VRF CE?

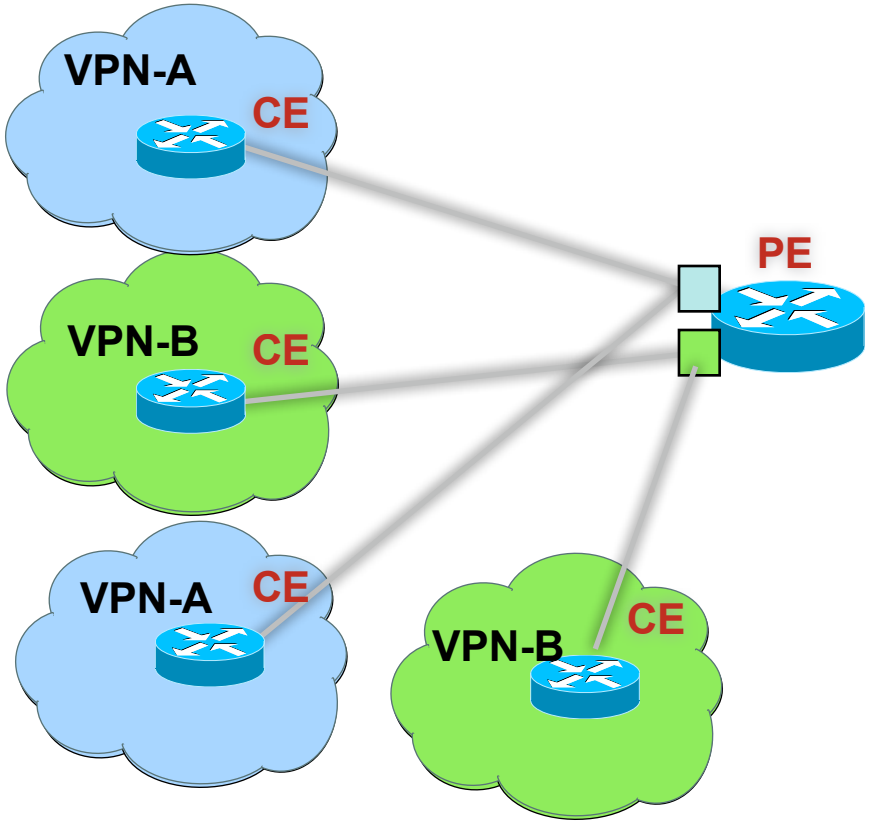
- Multi-VRF CE architecture uses the VRF concept to support multiple (overlapping and independent) routing tables (and forwarding tables) per customer
- Not a feature but an application based on VRF implementation
- Any routing protocol supported by normal VRF can be used in a Multi-VRF CE implementation
- The CE supports traffic separation between customer networks
- There is no MPLS functionality on the CE, no label exchange between the CE and PE

What is Multi-VRF CE



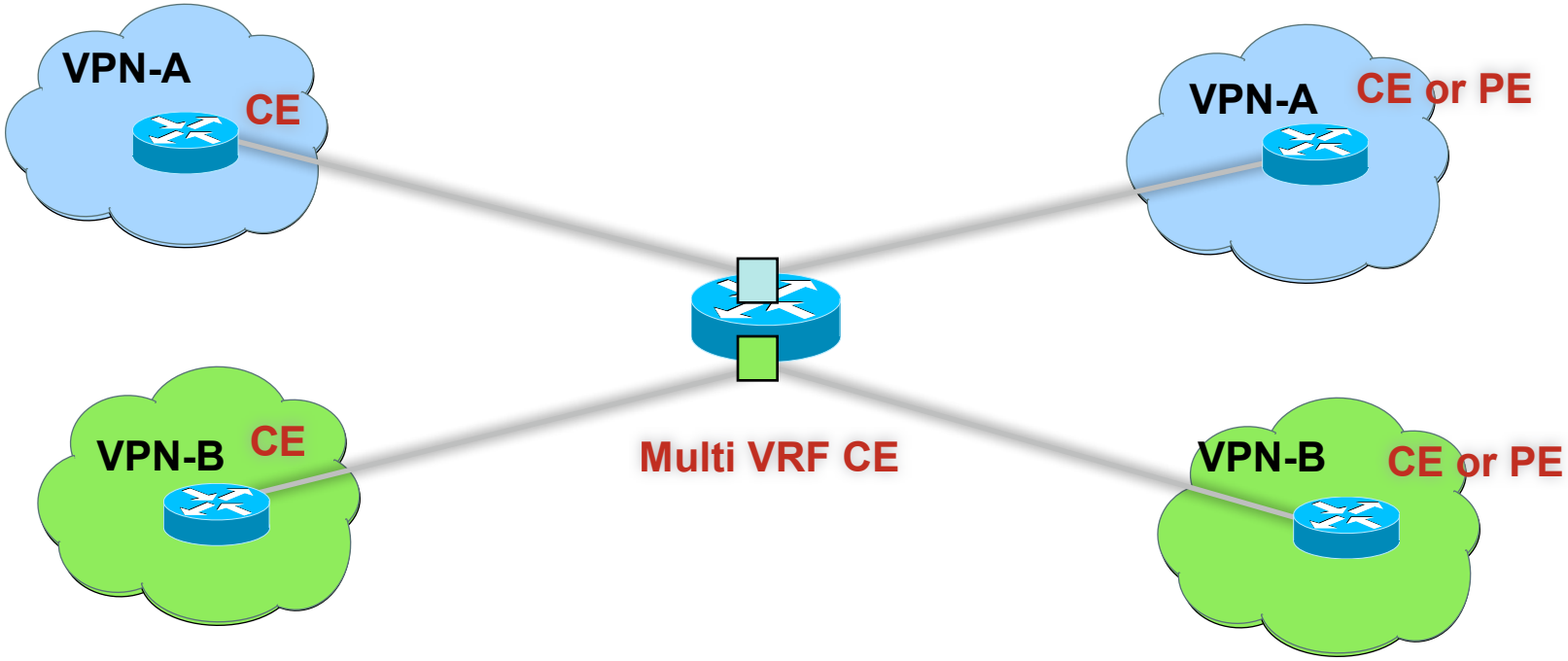
Take the existing PE VRF Functionality…

What is Multi-VRF CE



…And Remove the MPLS cloud

What is Multi-VRF CE

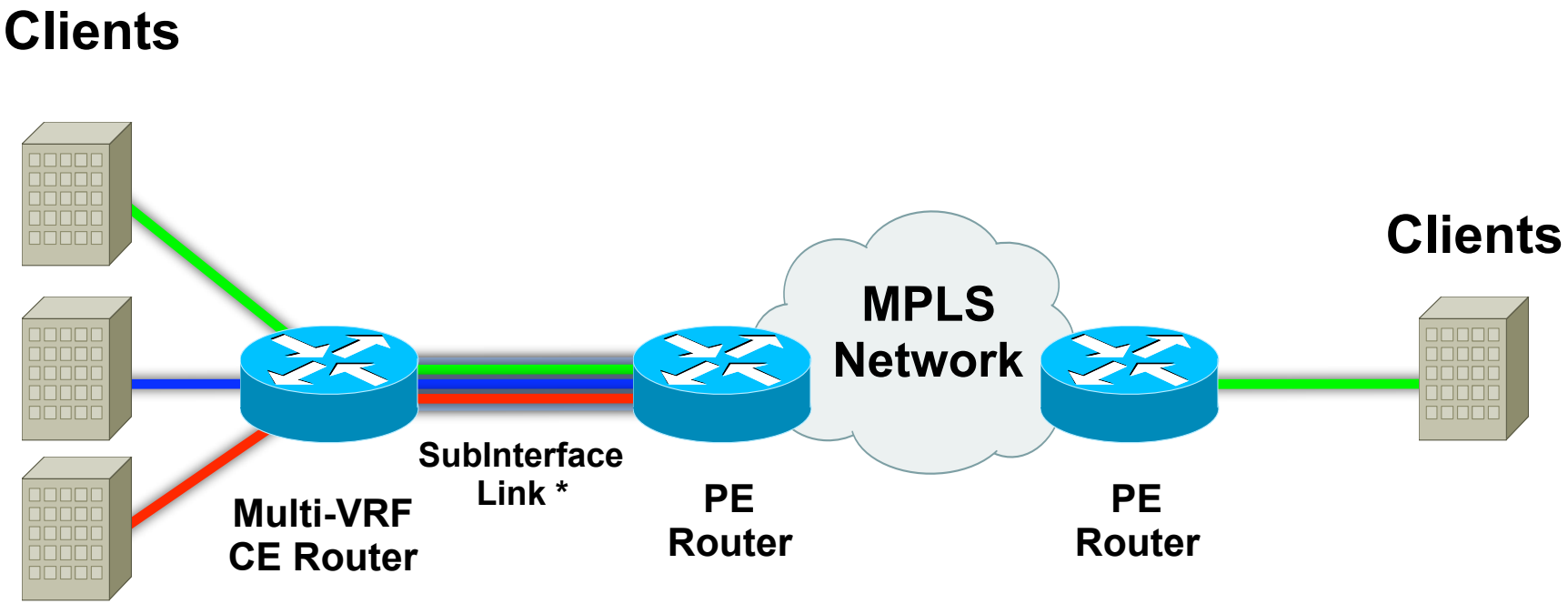


Put it at the customer site and call it a Multi-VRF CE

Applications: Two Examples

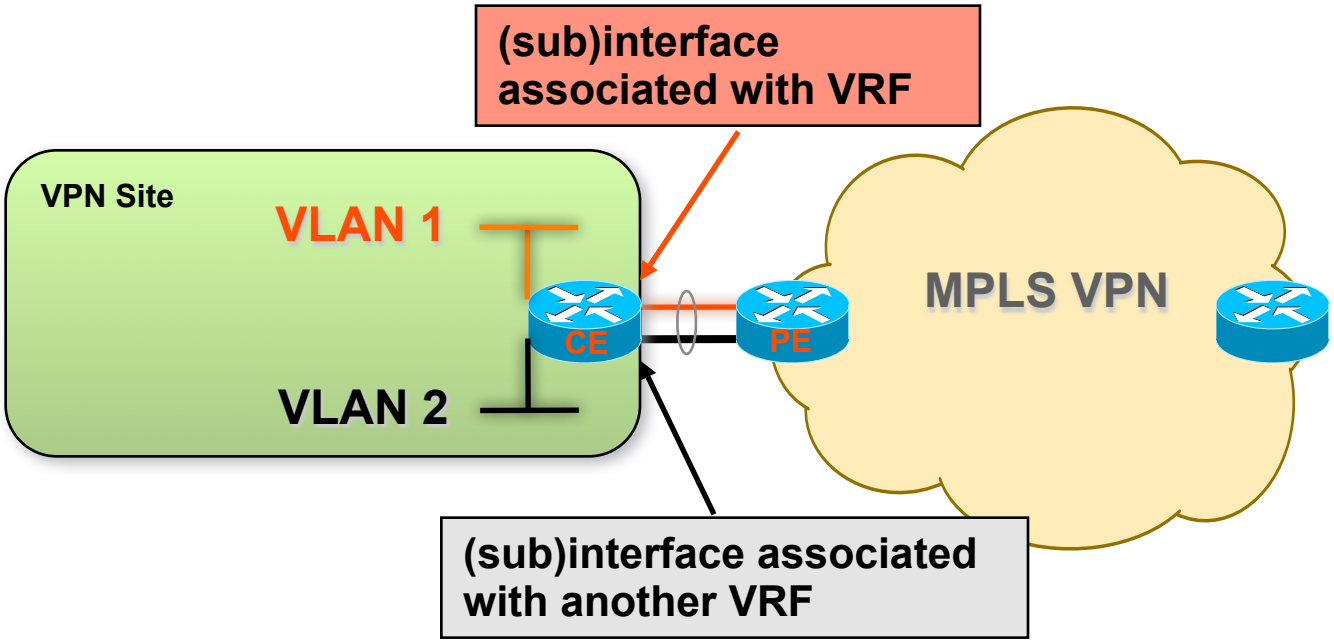
- Implement Multiple VPNs in a customer site using a single router
- Internet and VPN Service Using the Same CE – solution is attractive for small businesses that do not want to install separate CE routers for each service

Multi-VRF CE - Extending MPLS-VPN



Sub-Interface Link – Any Interface type that supports Sub Interfaces, FE-VLAN, Frame Relay, ATM VC's

Multi-VRF CE - a standalone Virtual-router !



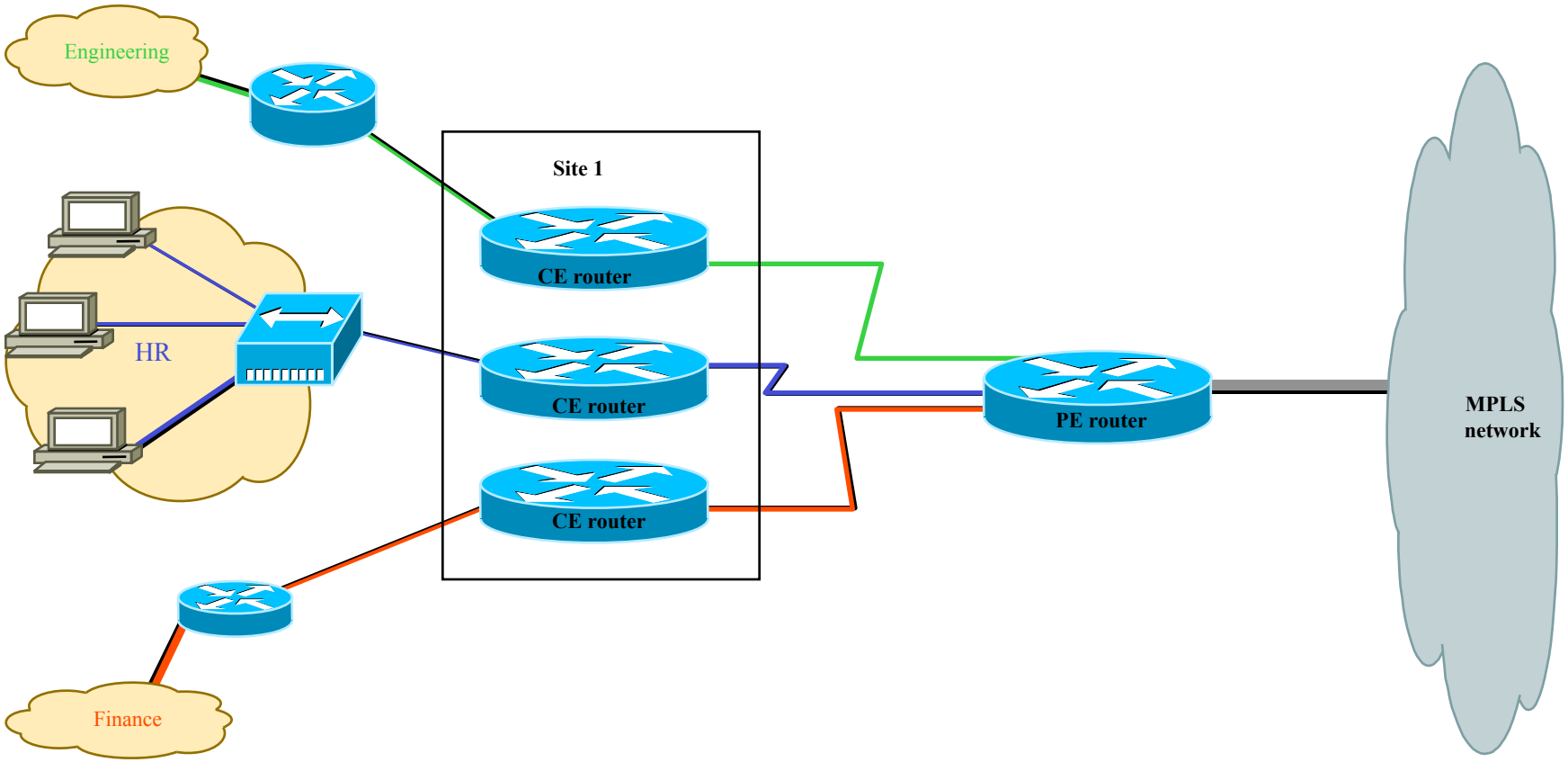
No MPLS, nor MP-iBGP on CE

Local Inter-VRF routing is supported

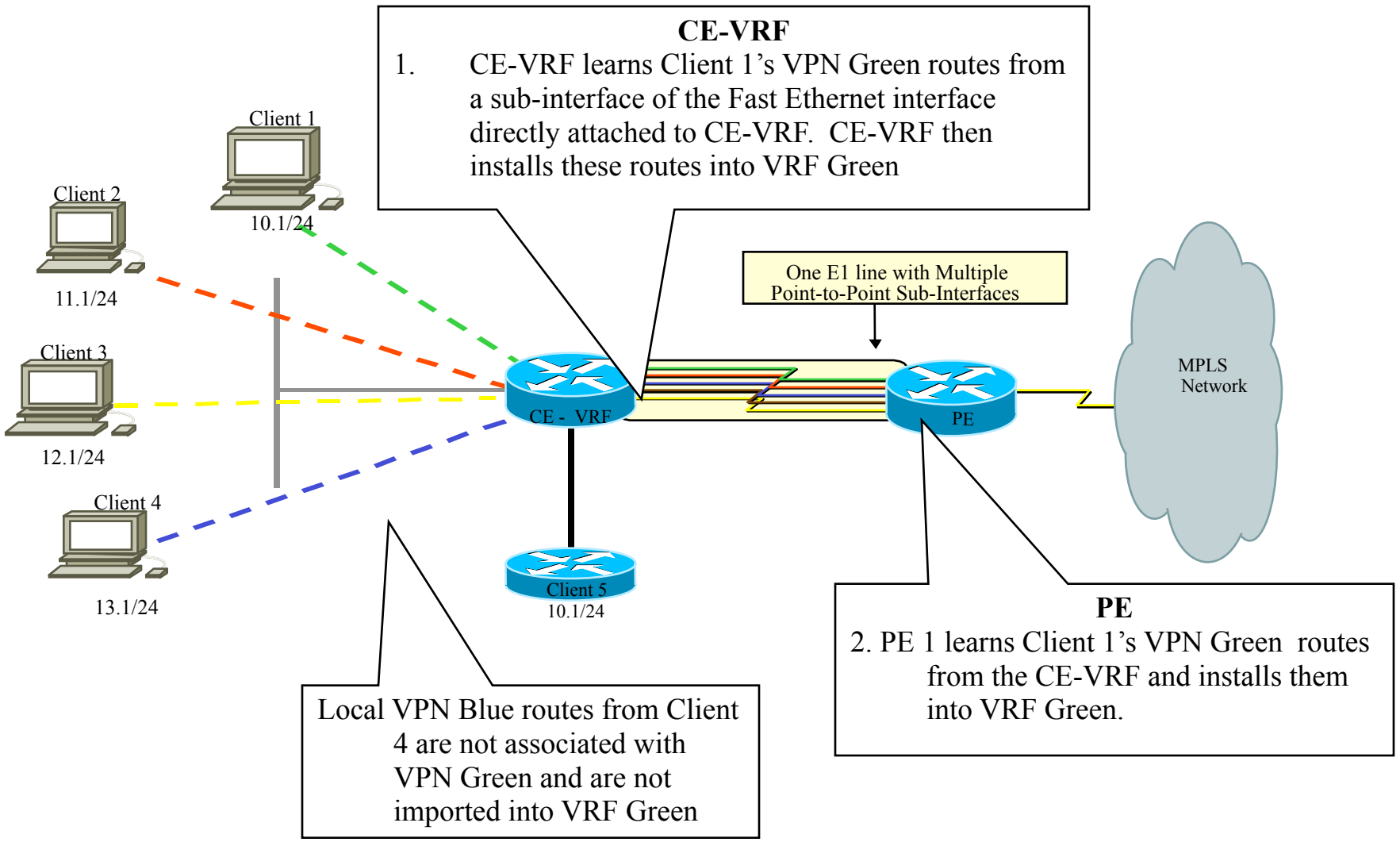
Multi-VRF CE Architecture

- Enhanced branch office capability
- CE routers use VRF interfaces VLAN-like configuration on the customer side
- CE router can only configure VRF interfaces and support VRF routing tables
- Use using a Multi-vrf CE is an alternative to separate CE routers per each client's organization

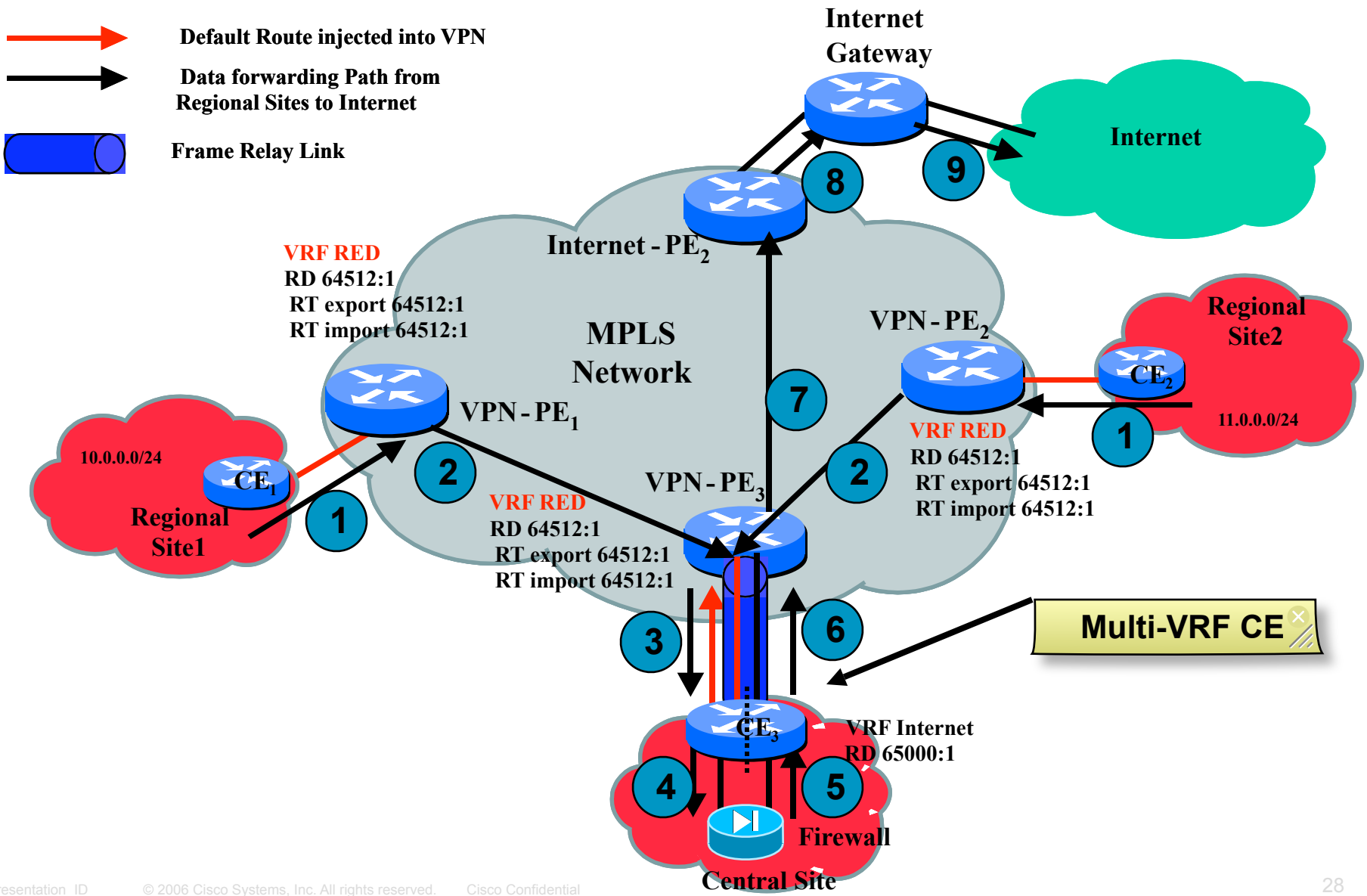
Multi-VRF CE Architecture: Replaces Separate CE Routers



Multi-VRF CE Architecture: Operational Model



Application 1: Internet Services and VPN Services Using A Single CE



Application 2: Multiple VPNs in a Customer Site Using a Single Router

- Objective: Provide building connectivity via Multi-VRF CE. Multiple departments or companies sharing a building need to be isolated from each other (e.g. financial departments).

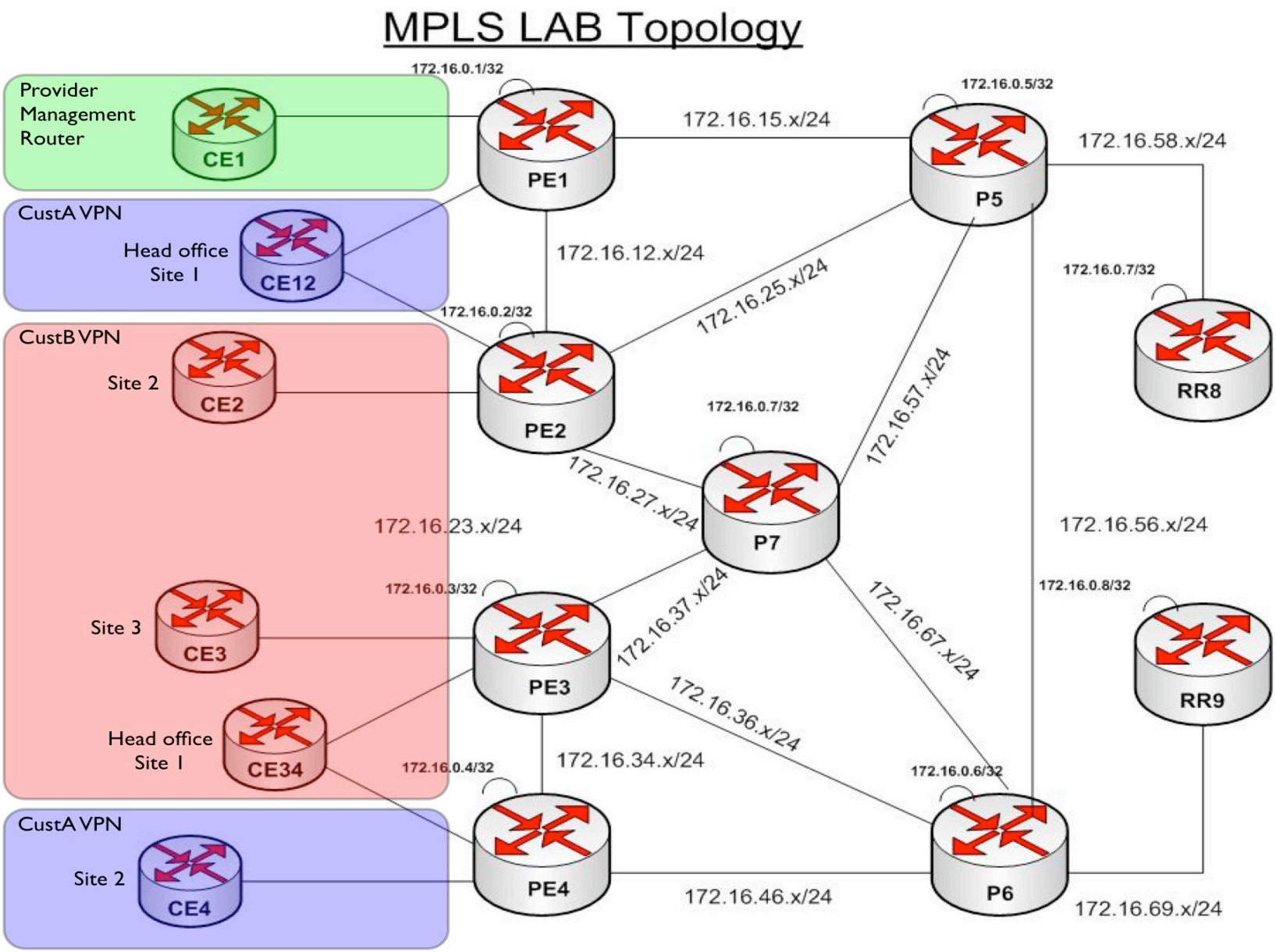
Conclusions

- Multi-VRF/VRF-Lite offers the following benefits:
  - Only one CE router is needed facilitating provisioning and network management rather than a multiple CE router solution
  - CE router has VRF functionality without full PE functionality to provide BGP routing tables
    - Note scalability factors
  - Less routing updates to manage
  - Overlapping Customer address spaces
  - Can co-exist with an MPLS-based network but no MPLS enabled on CE
    - Note applicability example for branch offices with multiple networks

MPLS Bootcamp

Management VRF

Management VRF aka Managed Services



Management VRF aka Managed Services

VRF name	Export RT	Import RT
CustA-VPN	100:1 and [Export map 100:6500 will match the connected routes)	100:1 and 100:6501
CustB-VPN	100:2 and [Export map 100:6500 will match the connected routes)	100:2 and 100:6501
VRF-Mgmt	100:6501 (only connected interface or route to NMS server)	100:6500

LDP Label Filtering

LDP Label Binding Filtering

MPLS LDP Outbound Label Filtering

Ability to filter local binding assignment and distribution to LDP peers using ACLs

MPLS LDP Inbound Label Binding Filtering

Ability to filter remote binding updates from LDP peers using ACLs

Local Label Allocation Filtering

Ability to filter local (RIB) prefixes for label binding assignment and distribution to remote LDP peers using prefix lists

Local Label Allocation Filtering (LLAF)

- Ability to filter local (RIB) prefixes for label binding are assigned and distributed to remote LDP peers using prefix lists

  Filtering using prefix-lists or explicit host-routes filter command

- Selective label allocation to host routes only is also possible
- Key Benefits

  Reduces locally stored label bindings (in LIB) and number of label binding updates send to adjacent LDP neighbors

Filtering via prefix-list:

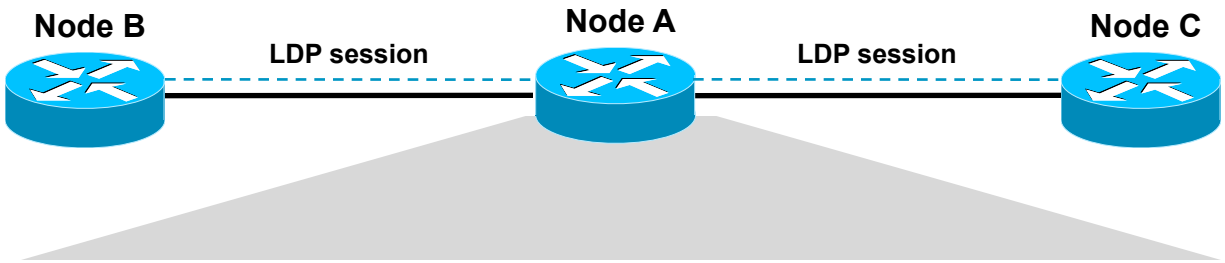
```
ip prefix-list list1 permit
  192.168.0.0/16 ge 18

mpls ldp label
  allocate global prefix-list list1
end
```

Filtering of host routes:

```
mpls ldp label
  allocate global host-routes
end
```

Example: LLAF for Host Routes

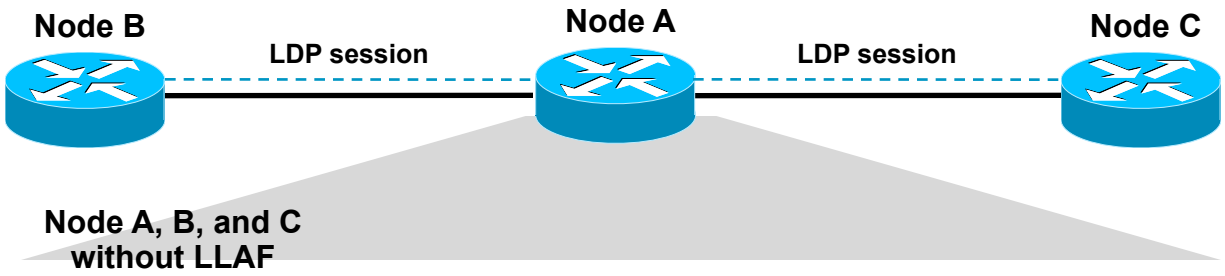
Example: LLAF for Host Routes



Node A:
Label Bindings stored in LIB

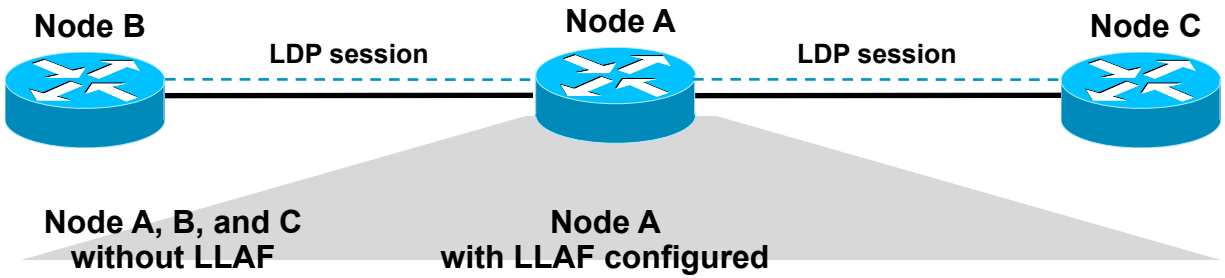
Node A:
Label Bindings exchanged with remote LDP peers

Example: LLAF for Host Routes



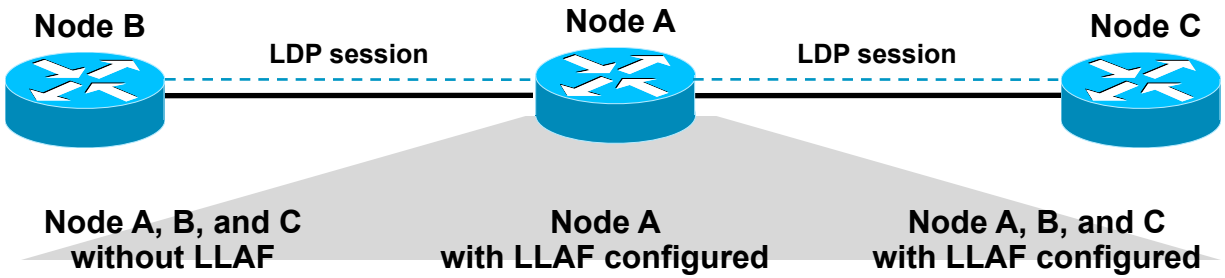
Node A: Label Bindings stored in LIB	• **For each IP prefix in RIB:** – Local label (node A) – Remote label (node B) – Remote label (node C)		
Node A: Label Bindings exchanged with remote LDP peers	• **Node A sends to B and C:** – (Local) label binding for each IP prefix in RIB: • **Node A receives from B/ C:** – Remote label binding for each IP prefix in RIB		

Example: LLAF for Host Routes



	Node A, B, and C without LLAF	Node A with LLAF configured	
Node A: Label Bindings stored in LIB	• **For each IP prefix in RIB:** – Local label (node A) – Remote label (node B) – Remote label (node C)	• **For host prefixes in RIB:** – Local label (node A) – Remote label (node B) – Remote label (node C) • **For non-host prefixes in RIB:** – **No Local label (node A)** – Remote label (node B) – Remote label (node C)	
Node A: Label Bindings exchanged with remote LDP peers	• **Node A sends to B and C:** – (Local) label binding for each IP prefix in RIB: • **Node A receives from B/ C:** – Remote label binding for each IP prefix in RIB	• **Node A sends to B and C:** – **(Local) label binding for only host prefixes** • **Node A receives from B/ C:** – **Remote label binding for each IP prefix in RIB**	

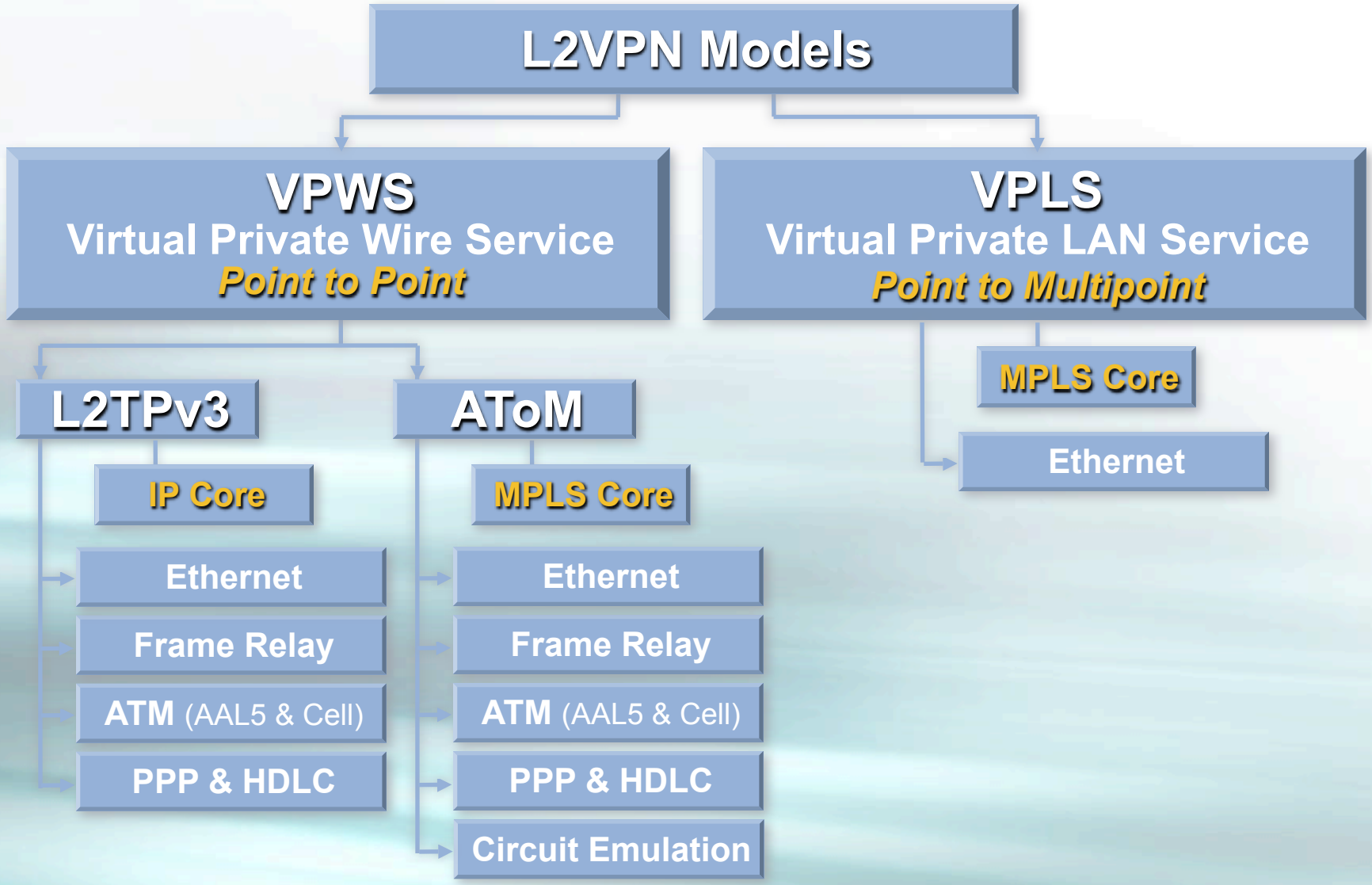
Example: LLAF for Host Routes



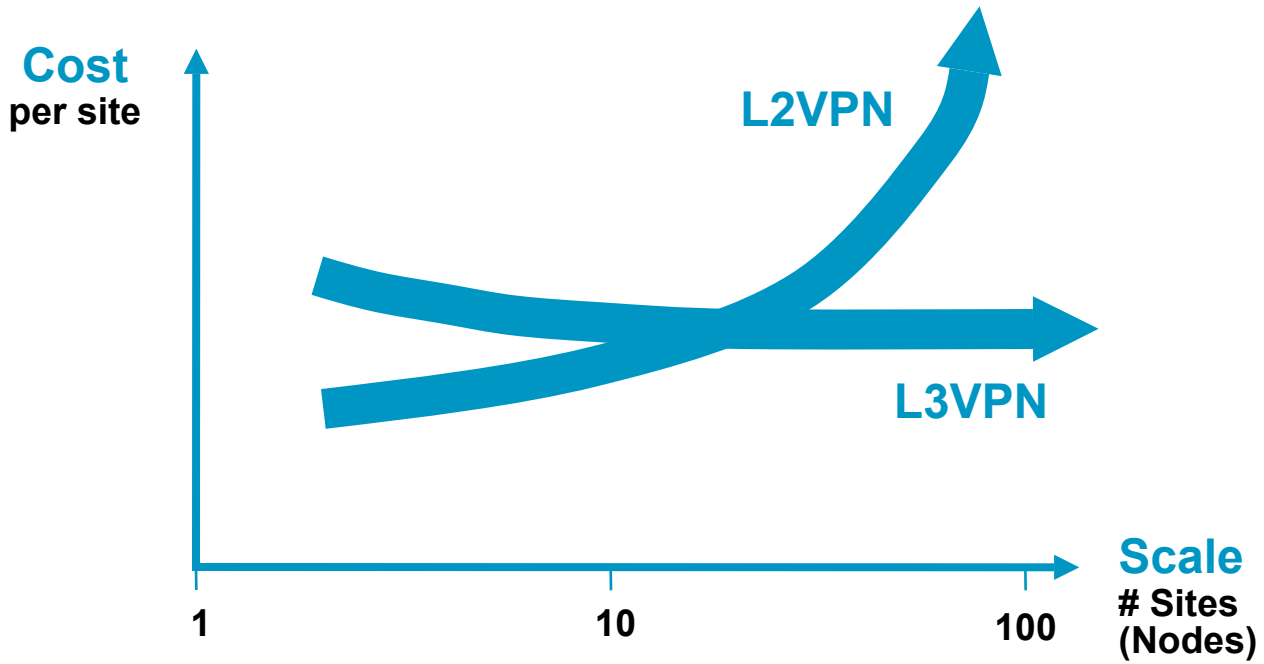
	Node A, B, and C without LLAF	Node A with LLAF configured	Node A, B, and C with LLAF configured
Node A: Label Bindings stored in LIB	• **For each IP prefix in RIB:** – Local label (node A) – Remote label (node B) – Remote label (node C)	• **For host prefixes in RIB:** – Local label (node A) – Remote label (node B) – Remote label (node C) • **For non-host prefixes in RIB:** – **No Local label (node A)** – Remote label (node B) – Remote label (node C)	• **For host prefixes in RIB:** – Local label (node A) – Remote label (node B) – Remote label (node C) • **No label bindings for any non-host prefixes in RIB**
Node A: Label Bindings exchanged with remote LDP peers	• **Node A sends to B and C:** – (Local) label binding for each IP prefix in RIB: • **Node A receives from B/ C:** – Remote label binding for each IP prefix in RIB	• **Node A sends to B and C:** – **(Local) label binding for only host prefixes** • **Node A receives from B/ C:** – **Remote label binding for each IP prefix in RIB**	• **Node A sends to B and C:** – **(Local) label binding for only host prefixes** • **Node A receives from B/ C:** – **Label binding for only host prefixes**

MPLS L2VPNs

Layer 2 VPN Taxonomy



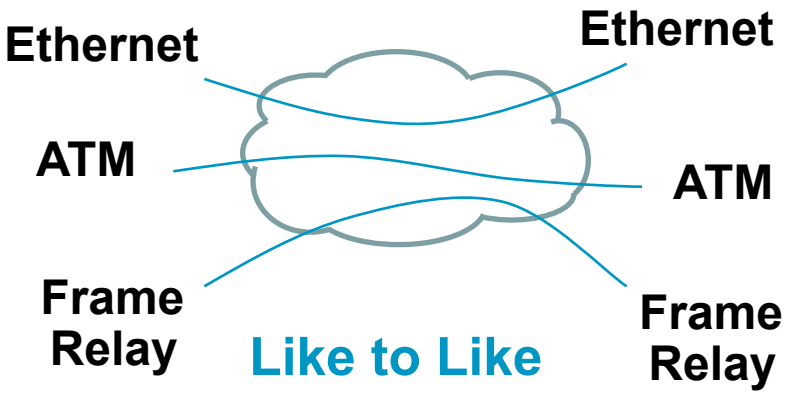
When L2VPN, When L3VPN?

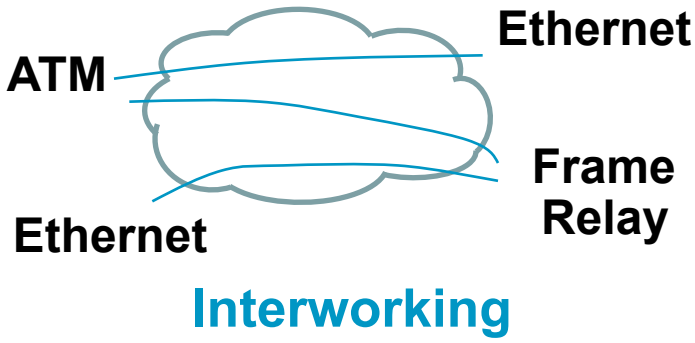


- As number of sites increases,
  - L2VPN: Cost of maintaining mesh connections increases
  - L3VPN: Cost of provisioning additional sites is lower

Technology for L2VPNs: Pseudowires

- L2VPNs are built with "Pseudowire" (PW) technology
- PWs provide an common intermediate format to transport multiple types of network services over a Packet Switched Network (PSN)
- PW technology provides Like-to-Like (L2L) transport and also Interworking (IW)





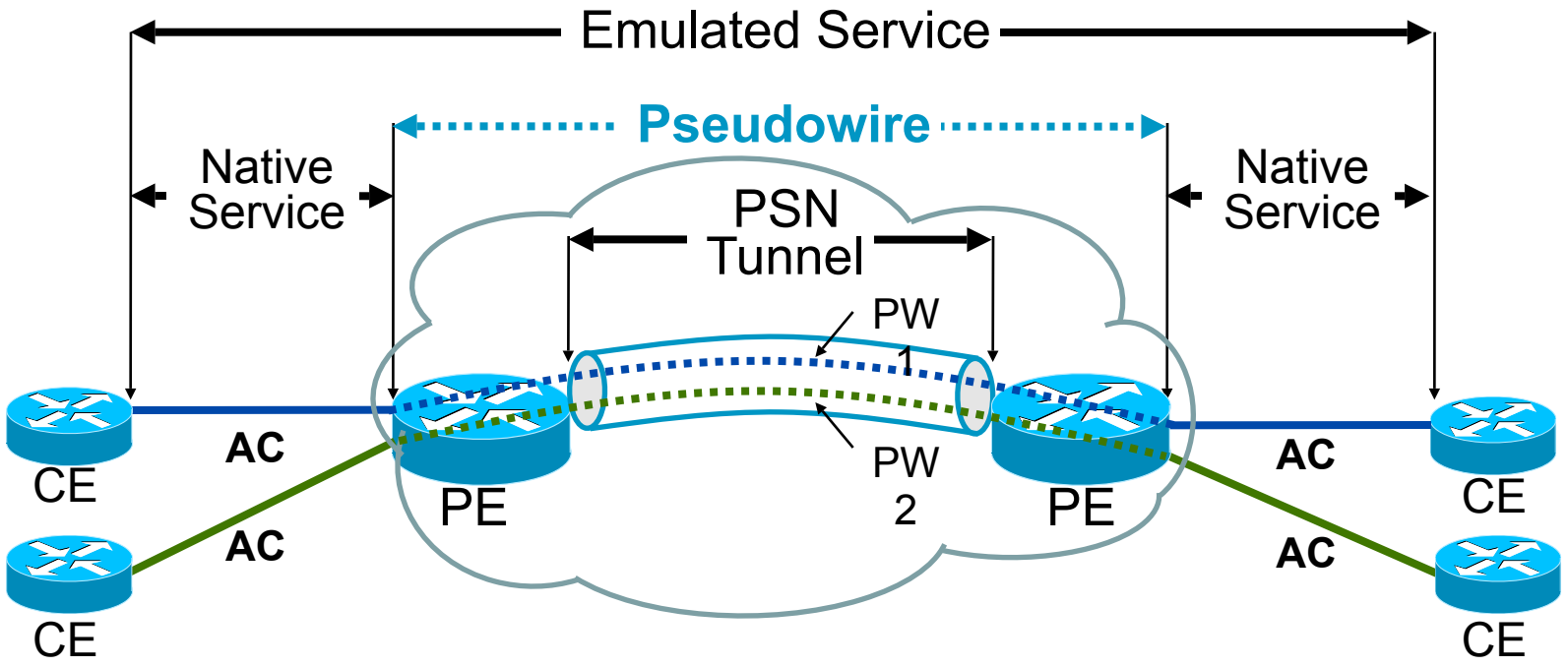
What Is a Pseudowire?

- A pseudowire (PW) is an emulation of a telecommunications service over a Packet Switched Network (PSN)
- PWs emulate the essential attributes of the native service

Payload Type	PW Service
Packet	Ethernet (all types), HDLC framing, PPP, Frame Relay, ATM AAL5 PDU
Cell	ATM
Bit stream	Unstructured E1/T1, E3/T3
Structured bit stream	SONET/SDH

- The PSN may be IP or IP/MPLS

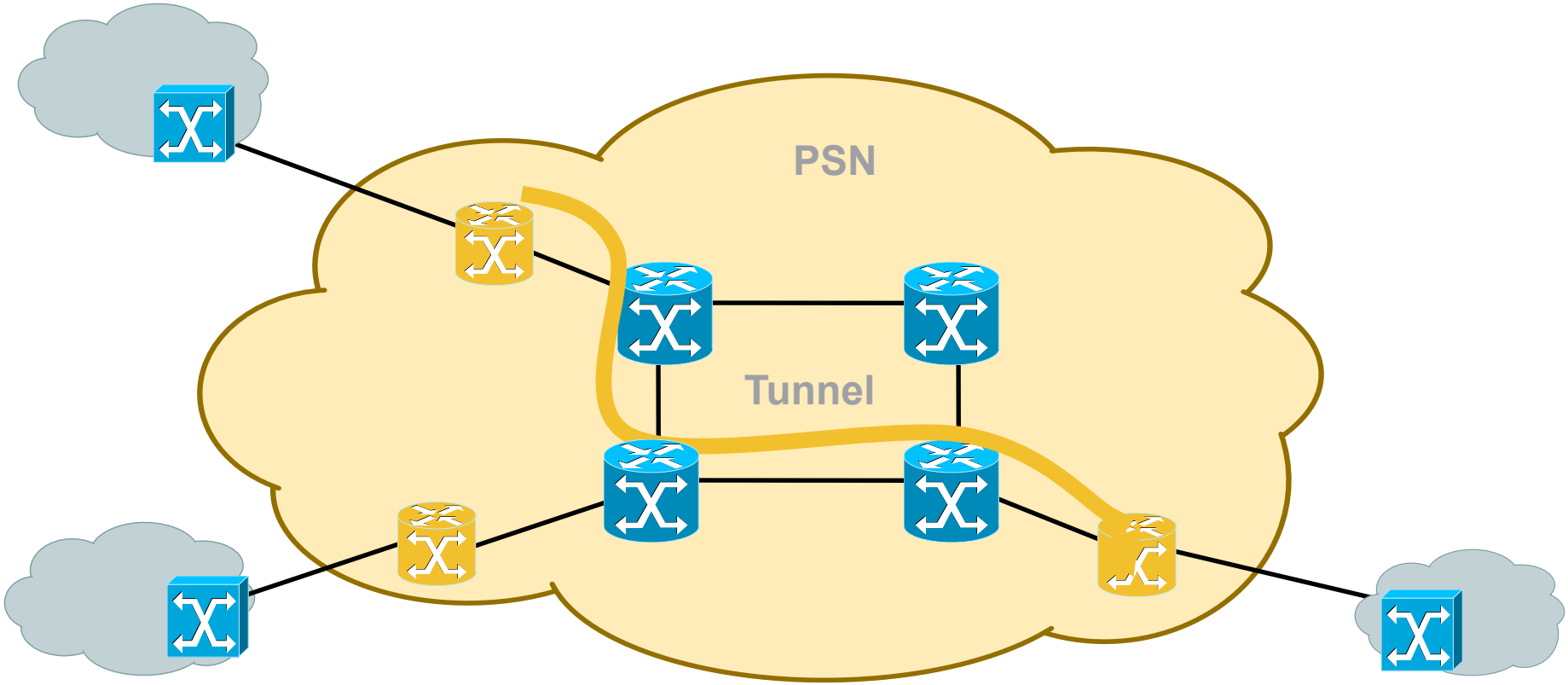
Pseudowire Reference Model



- 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

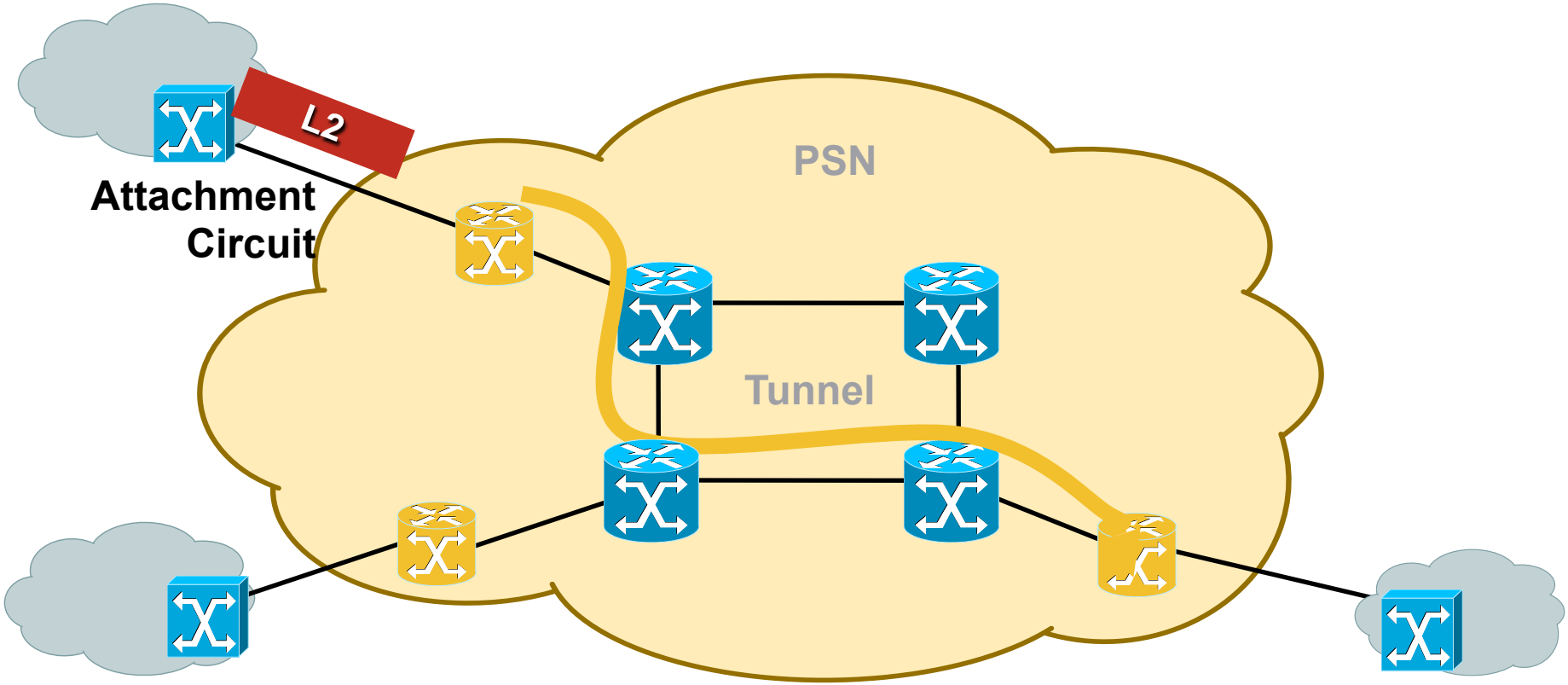
Ref: RFC 3985 Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture, March 2005

Generic L2VPN Architecture



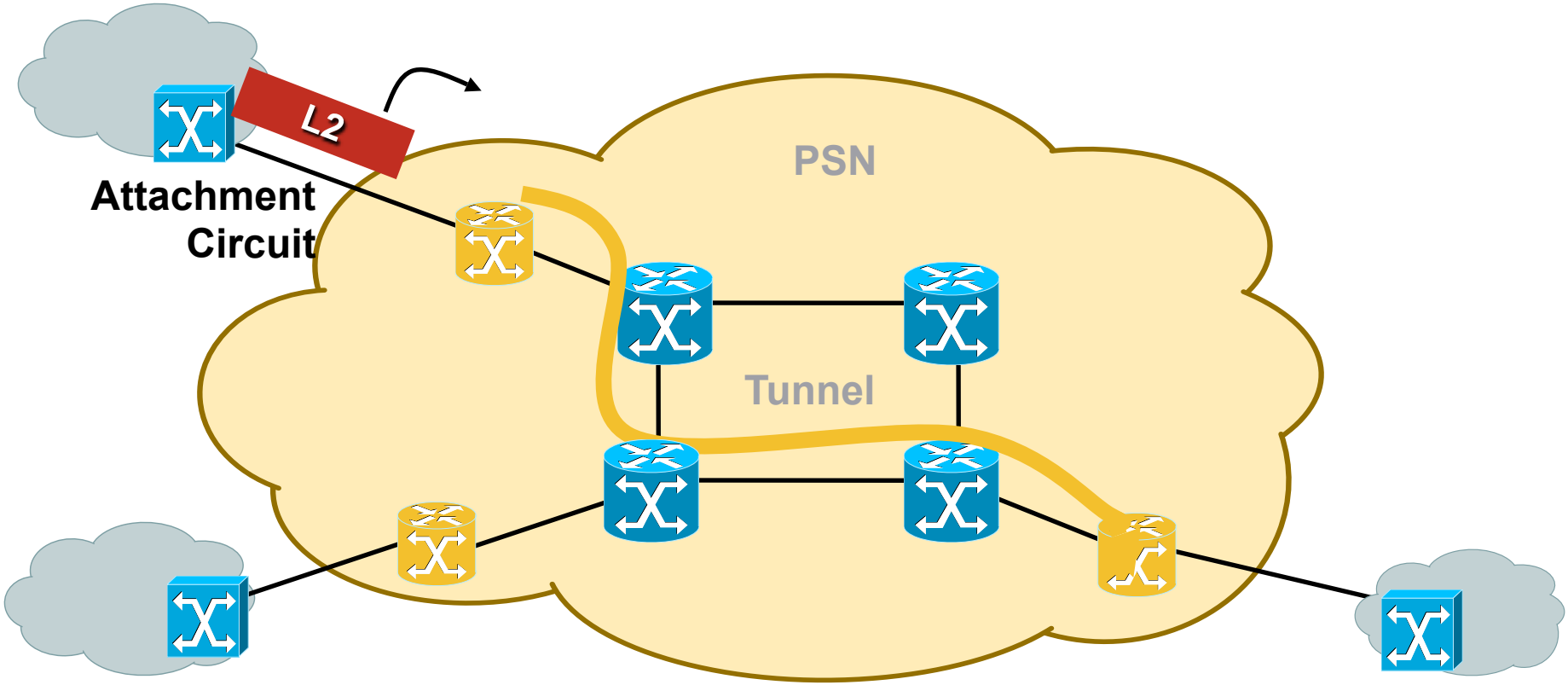
- Tunnels (MPLS, L2TPv3, GRE, IPSEC, etc.)
- Emulated VCs (pseudowires) inside tunnels (many-to-one)
- Attachment VCs (e.g. FR DLCI, PPP) mapped to emulated VCs

Generic L2VPN Architecture



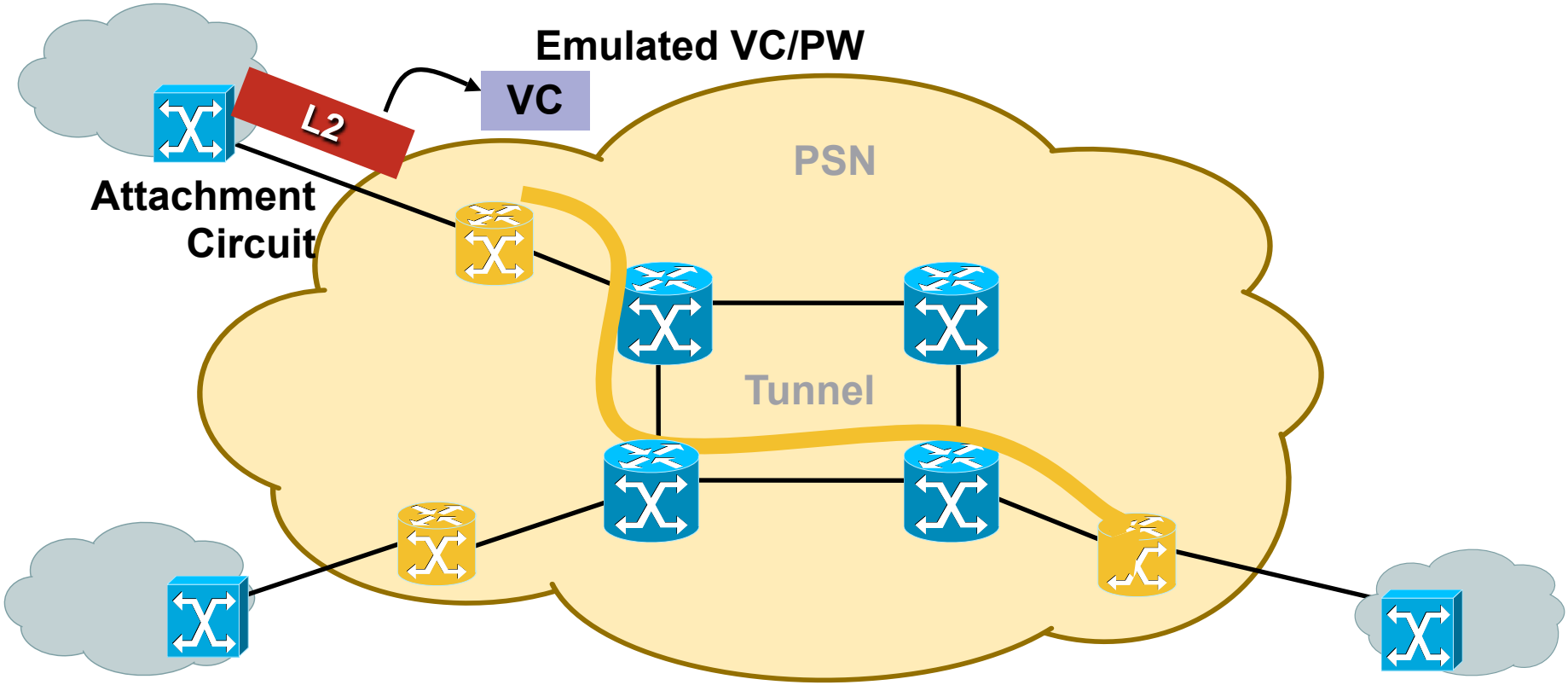
- Tunnels (MPLS, L2TPv3, GRE, IPSEC, etc.)
- Emulated VCs (pseudowires) inside tunnels (many-to-one)
- Attachment VCs (e.g. FR DLCI, PPP) mapped to emulated VCs

Generic L2VPN Architecture



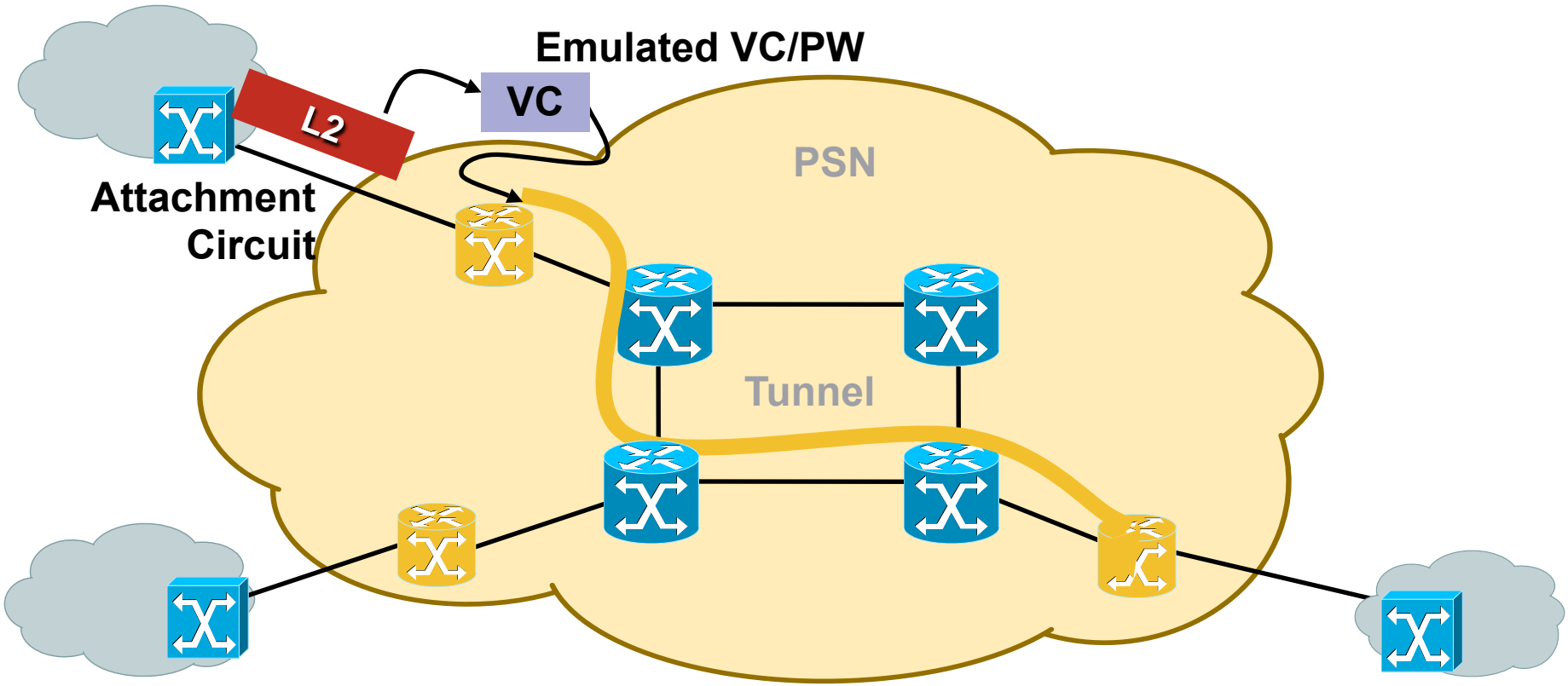
- Tunnels (MPLS, L2TPv3, GRE, IPSEC, etc.)
- Emulated VCs (pseudowires) inside tunnels (many-to-one)
- Attachment VCs (e.g. FR DLCI, PPP) mapped to emulated VCs

Generic L2VPN Architecture



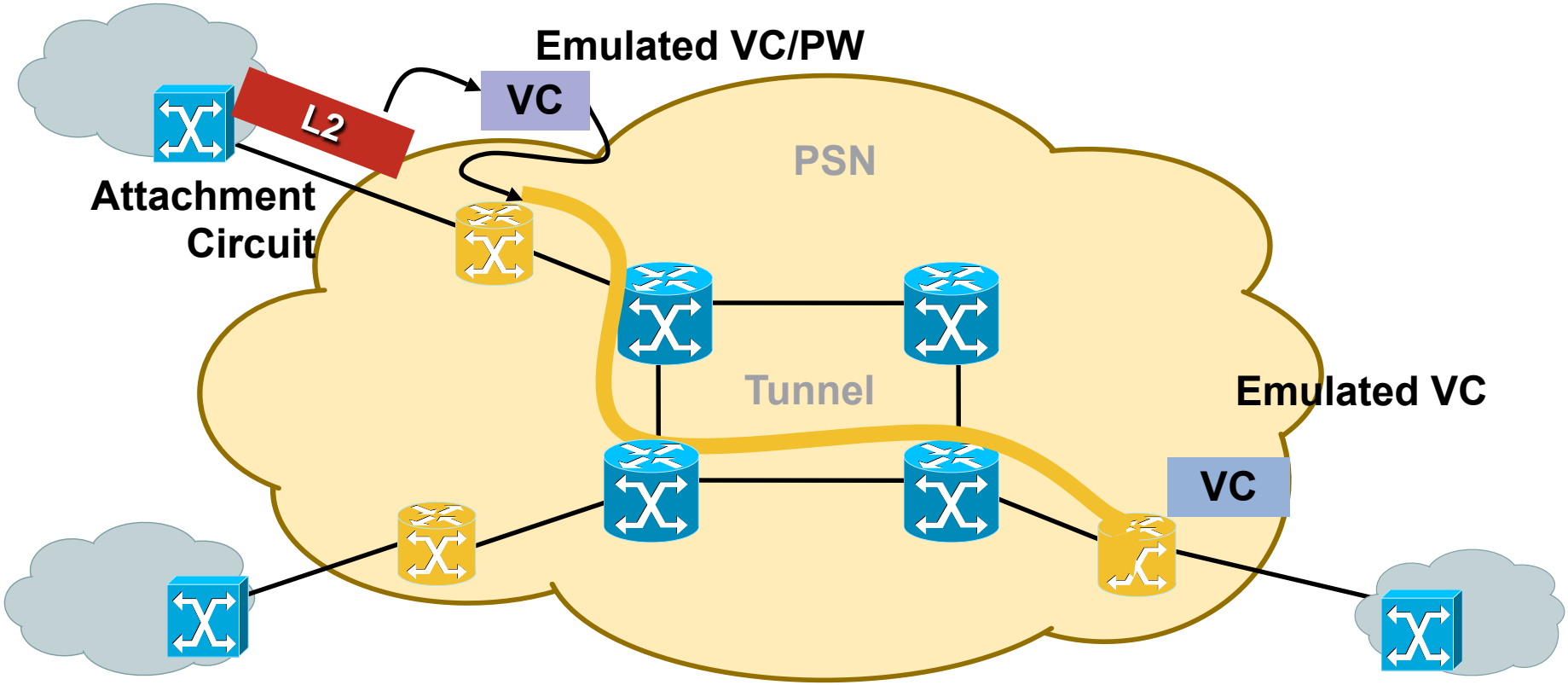
- Tunnels (MPLS, L2TPv3, GRE, IPSEC, etc.)
- Emulated VCs (pseudowires) inside tunnels (many-to-one)
- Attachment VCs (e.g. FR DLCI, PPP) mapped to emulated VCs

Generic L2VPN Architecture



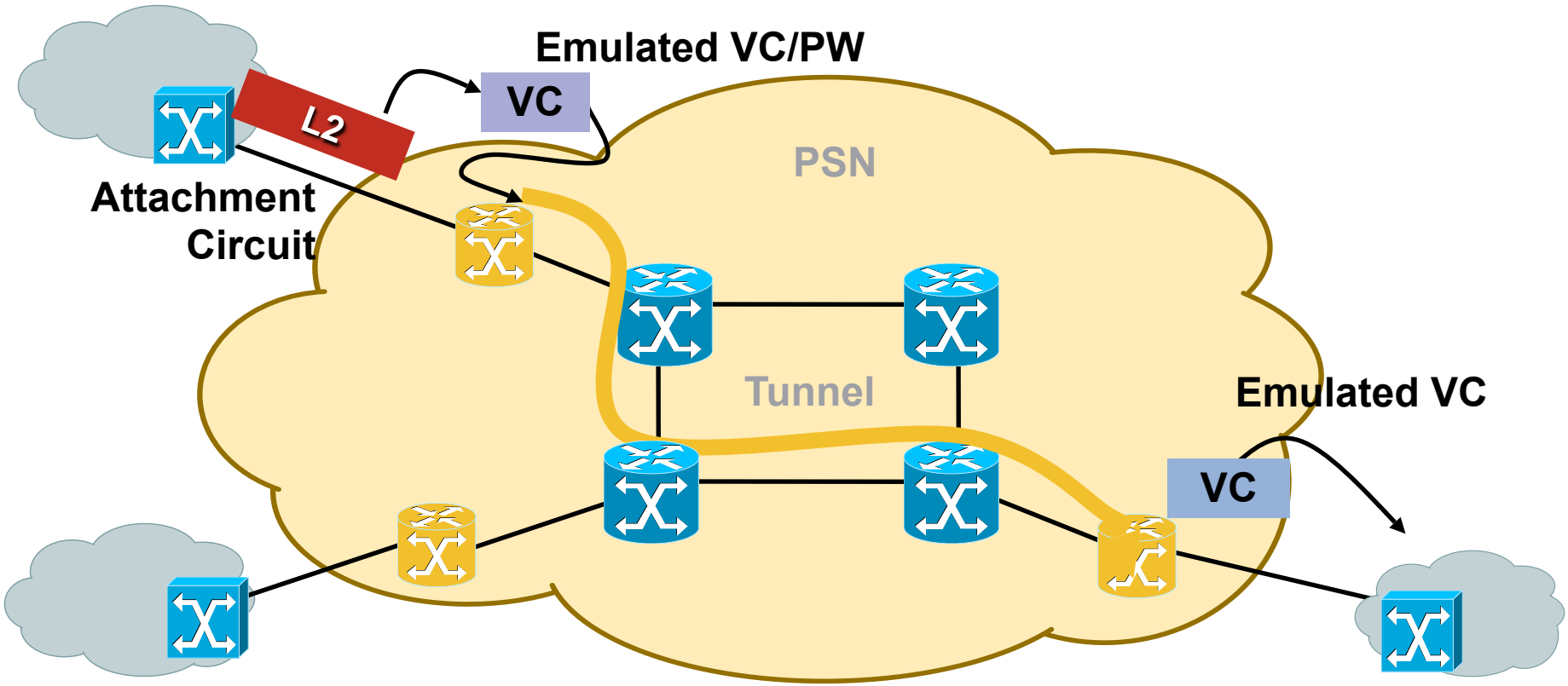
- Tunnels (MPLS, L2TPv3, GRE, IPSEC, etc.)
- Emulated VCs (pseudowires) inside tunnels (many-to-one)
- Attachment VCs (e.g. FR DLCI, PPP) mapped to emulated VCs

Generic L2VPN Architecture



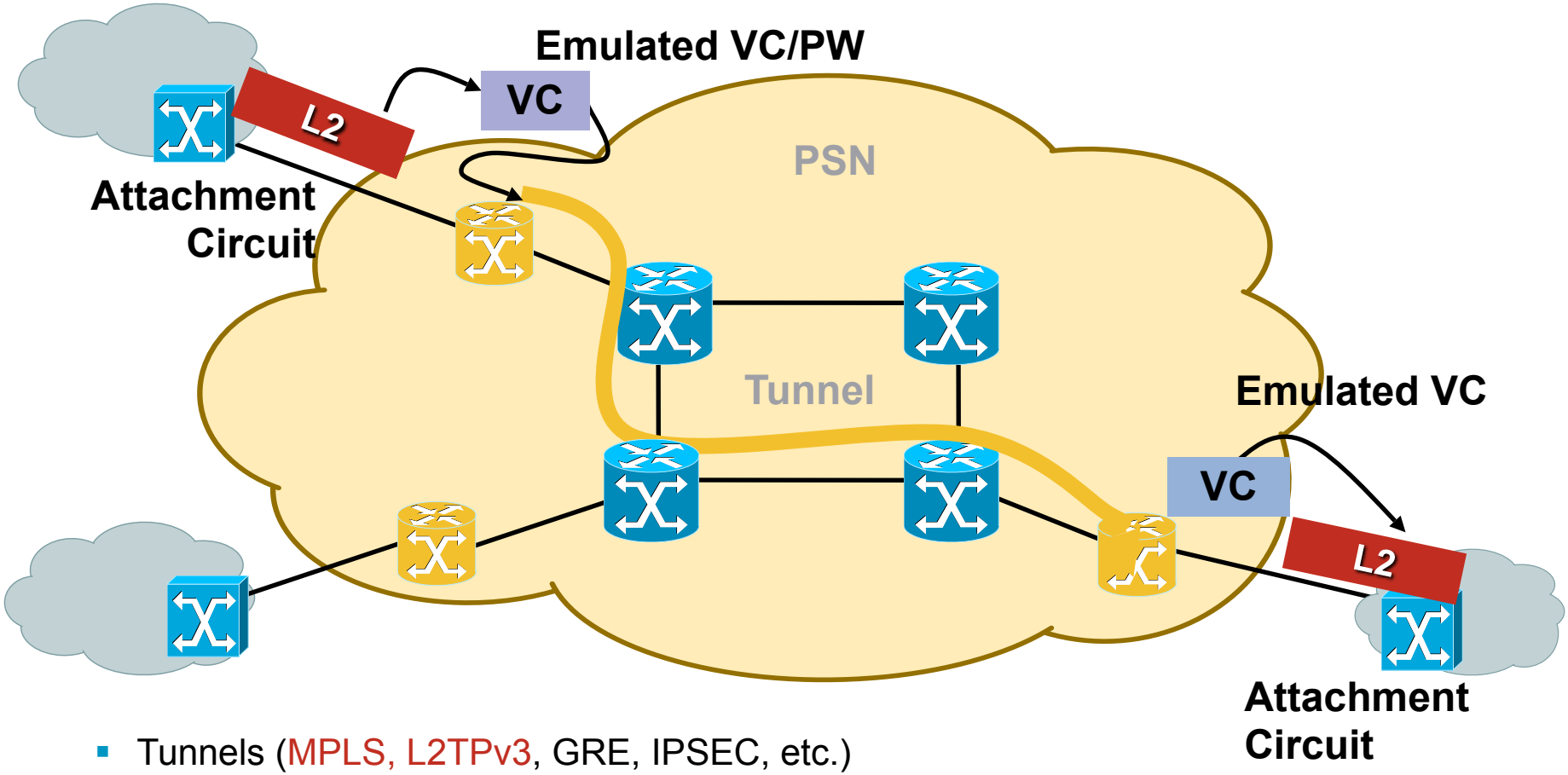
- Tunnels (MPLS, L2TPv3, GRE, IPSEC, etc.)
- Emulated VCs (pseudowires) inside tunnels (many-to-one)
- Attachment VCs (e.g. FR DLCI, PPP) mapped to emulated VCs

Generic L2VPN Architecture



- Tunnels (MPLS, L2TPv3, GRE, IPSEC, etc.)
- Emulated VCs (pseudowires) inside tunnels (many-to-one)
- Attachment VCs (e.g. FR DLCI, PPP) mapped to emulated VCs

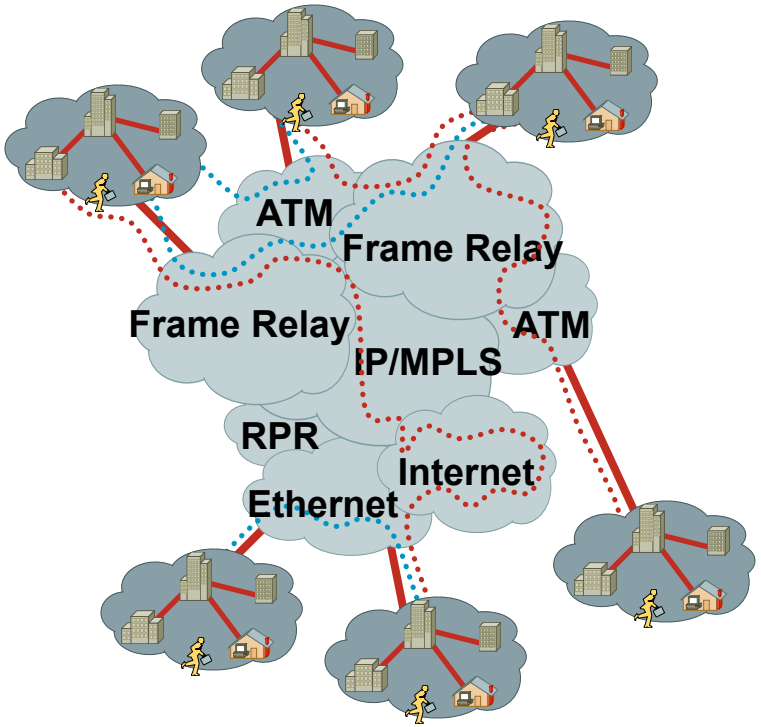
Generic L2VPN Architecture

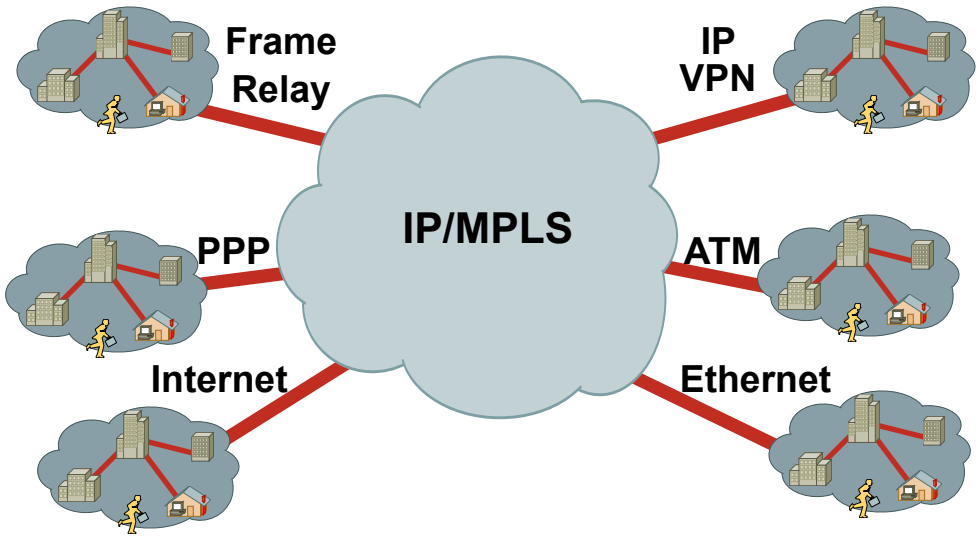


- Tunnels (MPLS, L2TPv3, GRE, IPSEC, etc.)
- Emulated VCs (pseudowires) inside tunnels (many-to-one)
- Attachment VCs (e.g. FR DLCI, PPP) mapped to emulated VCs

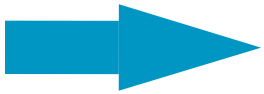
Any Transport over MPLS (AToM)

Pseudowire Technology: Enables Multiple Services over Common Infrastructure





Many Services,
Many Networks

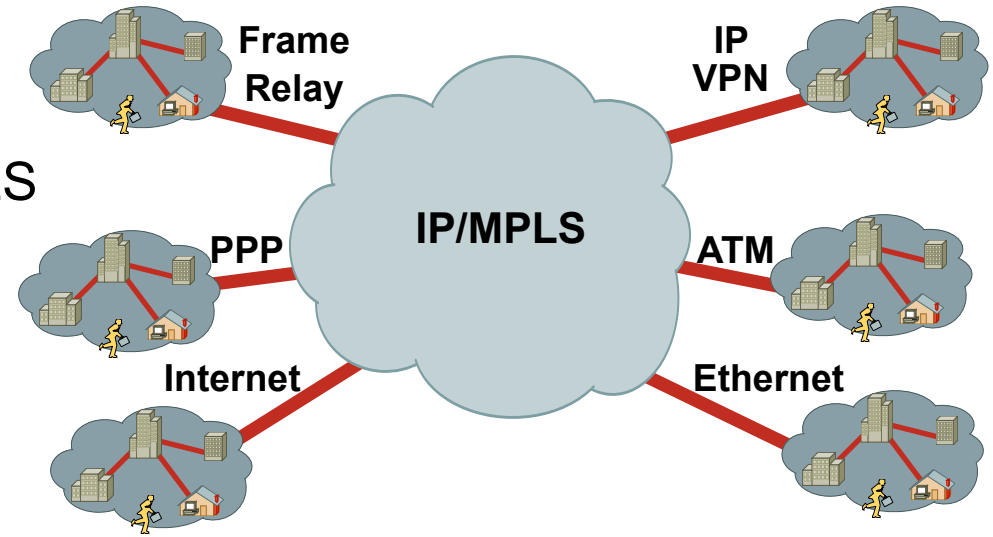


Many Services,
One Network

Drivers
Any Transport over MPLS (AToM)

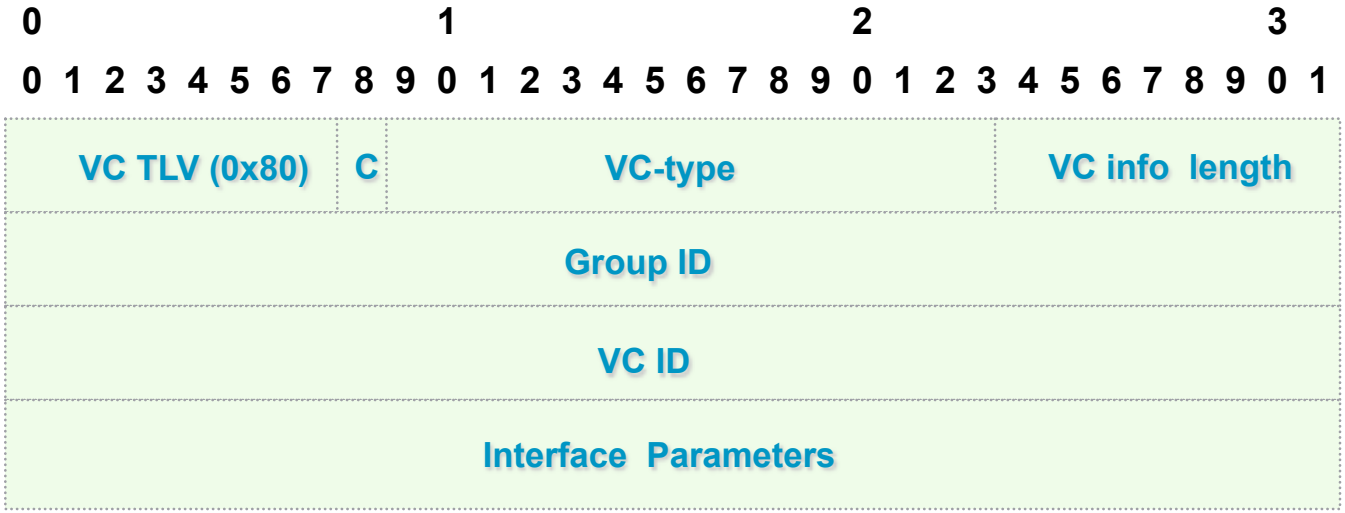
- Enables providers to carry multiple network services over a single, converged network using IP/MPLS
  - Ethernet over MPLS
  - Frame Relay over MPLS
  - ATM AAL5 over MPLS
  - ATM Cell Relay over MPLS
  - PPP over MPLS
  - HDLC over MPLS
  - TDM over MPLS

New



- **More services at lower costs and lower complexity**

AToM: Virtual Circuit FEC Element



C: Control Word (1 bit) – Control word present if bit set

VC-type (15 bits) - Type of VC e.g FR, ATM, VLAN, Ethernet, PPP, HDLC

VC info length (8 bits) – Length of VCID field and interface parameters

Group ID (32 bits) – Represents a groups of VCs. Can be used for mass label withdrawal

VC ID (32 bits) – Connection identifier used in conjunction with the VC-type to identify a particular VC

Interface Parameters (Variable) – Edge facing interface parameters, such as MTU

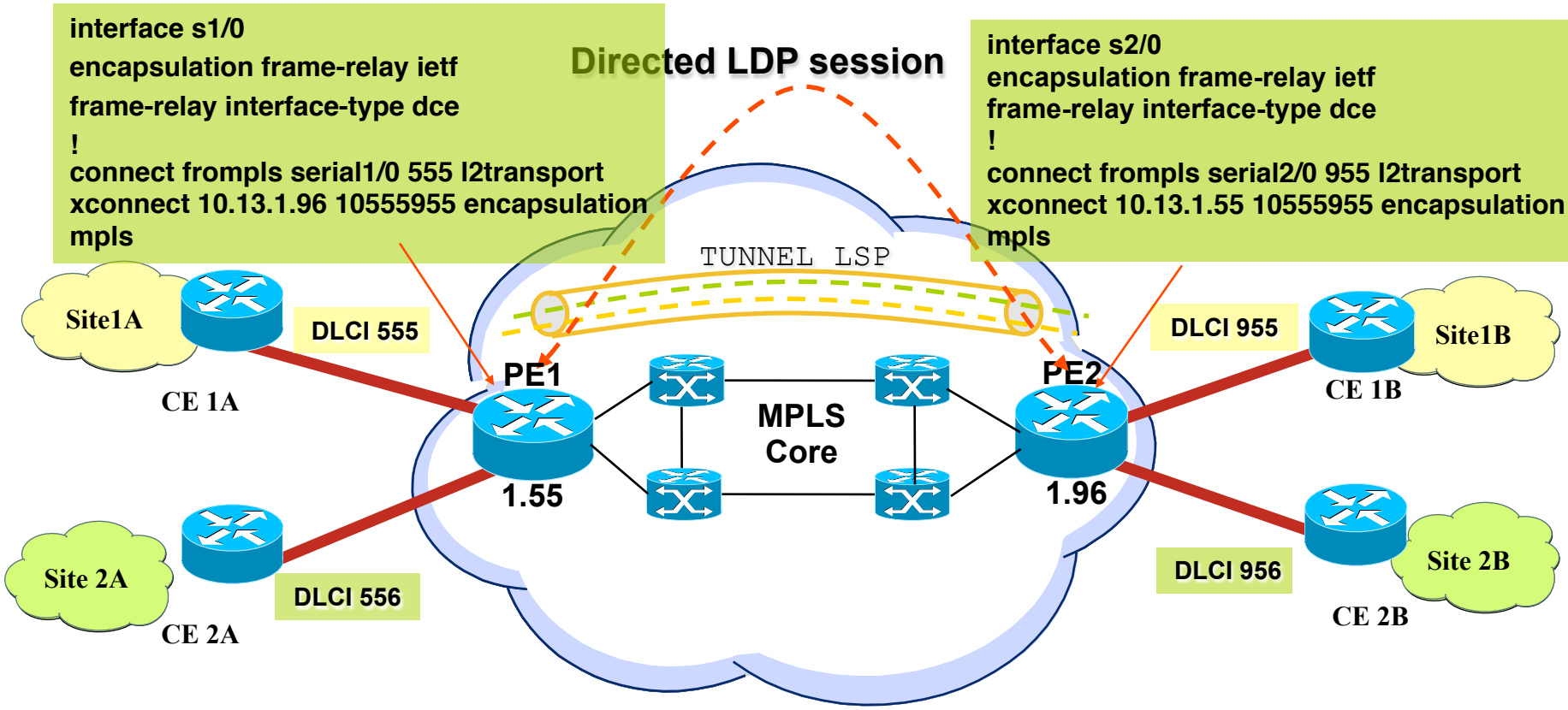
Courtesy: Jim Guichard

Some Currently Defined VC-Types

PW type	Description	
0x0001	**Frame Relay DLCI**	*! Frame Relay*
0x0002	**ATM AAL5 SDU VCC transport**	*! ATM AAL5 SDU*
0x0003	**ATM transparent cell transport**	*! ATM Cell Port Mode*
0x0004	**Ethernet Tagged Mode**	*! Ethernet VLAN*
0x0005	**Ethernet**	*! Ethernet*
0x0006	**HDLC**	*! HDLC*
0x0007	**PPP**	*! PPP*
0x0008	SONET/SDH Circuit Emulation Service Over MPLS (CEM) [Note1]	
0x0009	**ATM n-to-one VCC cell transport**	*! ATM Cell VC Mode*
0x000A	**ATM n-to-one VPC cell transport**	*! ATM Cell VP Mode*
0x000B	**IP Layer2 Transport**	*! Interworking IP*
0x000C	ATM one-to-one VCC Cell Mode	
0x000D	ATM one-to-one VPC Cell Mode	
0x000E	ATM AAL5 PDU VCC transport	
0x000F	Frame-Relay Port mode	
0x0010	SONET/SDH Circuit Emulation over Packet (CEP)	
0x0011	Structure-agnostic E1 over Packet (SAToP)	
0x0012	Structure-agnostic T1 (DS1) over Packet (SAToP)	
0x0013	Structure-agnostic E3 over Packet (SAToP)	
0x0014	Structure-agnostic T3 (DS3) over Packet (SAToP)	
0x0015	CESoPSN basic mode	
0x0016	TDMoIP basic mode	
0x0017	CESoPSN TDM with CAS	
0x0018	TDMoIP TDM with CAS	

Note 1: This PW Type Is Grandfathered for a Historical Protocol; the Recommended Standards-Track Protocol to Use Is CEP (PW Type 0x0010)

AToM: Control Plane Example



Step1: "xconnect 10.13.1.96 10555955 encapsulation mpls" added to PE1 with FRoMPLS enabled for serial1/0

Step2: Directed LDP session setup with 10.13.1.96 to exchange VC labels

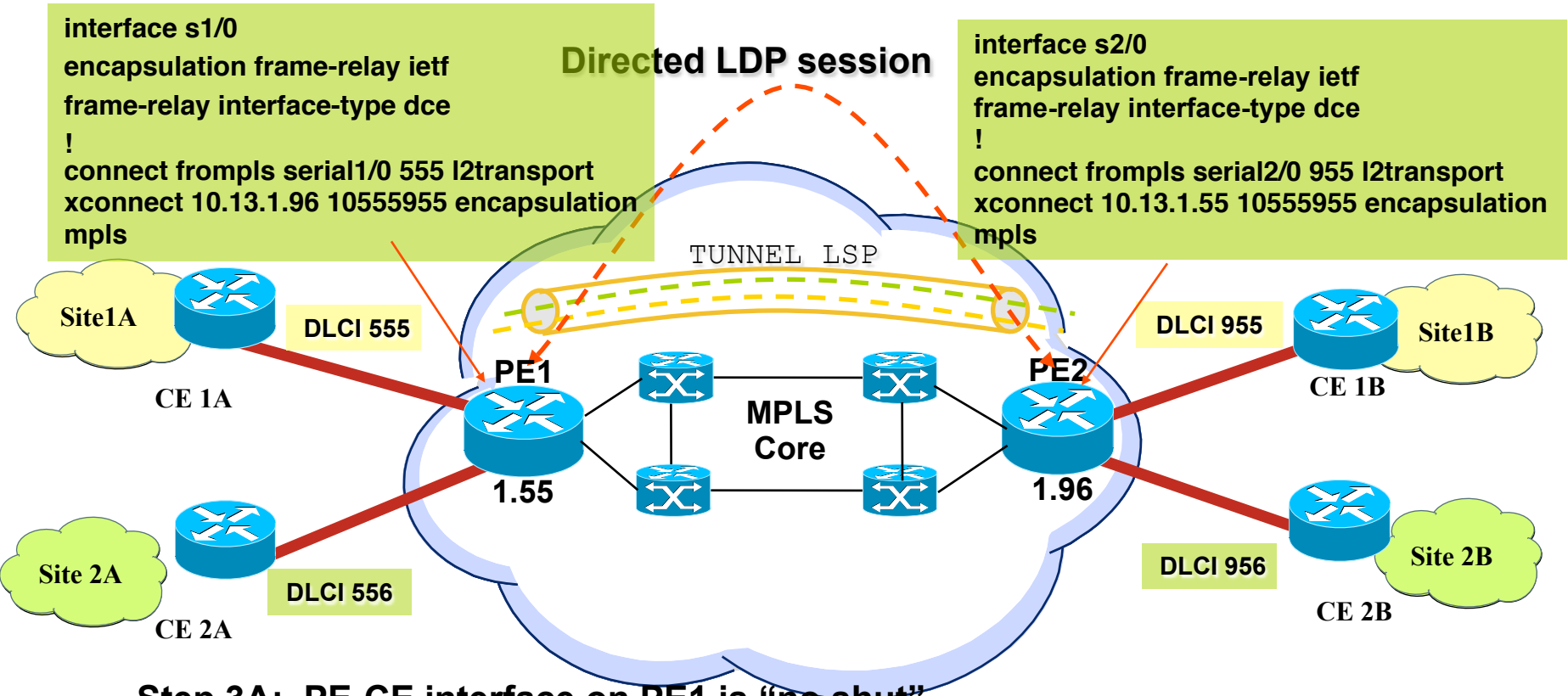
AToM: Discovery Phase

```
PE1#sh mpls ldp discovery detail
 Local LDP Identifier:
                10.13.1.55:0
 Discovery Sources:
       Interfaces:
              POS11/0/0 (tdp): xmit/recv
                  TDP Id: 10.13.1.58:0
                    Src IP addr: 10.13.5.41; Transport IP addr: 10.13.1.58
              FastEthernet10/0/0.441 (tdp): xmit/recv
                  TDP Id: 10.13.1.59:0
                    Src IP addr: 10.13.5.65; Transport IP addr: 10.13.1.59
              FastEthernet10/0/1.432 (tdp): xmit/recv
                  TDP Id: 10.13.1.58:0
                    Src IP addr: 10.13.5.61; Transport IP addr: 10.13.1.58
   Targeted Hellos:
                10.13.1.55 -> 10.13.1.96 (ldp): active/passive, xmit/recv
                LDP Id: 10.13.1.96:0
                        Src IP addr: 10.13.1.96; Transport IP addr:
   10.13.1.96
```

AToM: Targeted LDP session

```
PE1#sh mpls ldp neighbor 10.13.1.96
    Peer LDP Ident: 10.13.1.96:0; Local LDP Ident 10.13.1.55:0
        TCP connection: 10.13.1.96.11014 - 10.13.1.55.646
        State: Oper; Msgs sent/rcvd: 2773/2779; Downstream
        Up time: 1d10h
        LDP discovery sources:
          Targeted Hello 10.13.1.55 -> 10.13.1.96, active,
  passive
        Addresses bound to peer LDP Ident:
          10.13.1.96      10.13.9.30      10.13.9.46
  10.13.0.96
          10.13.9.66
PE1#
```

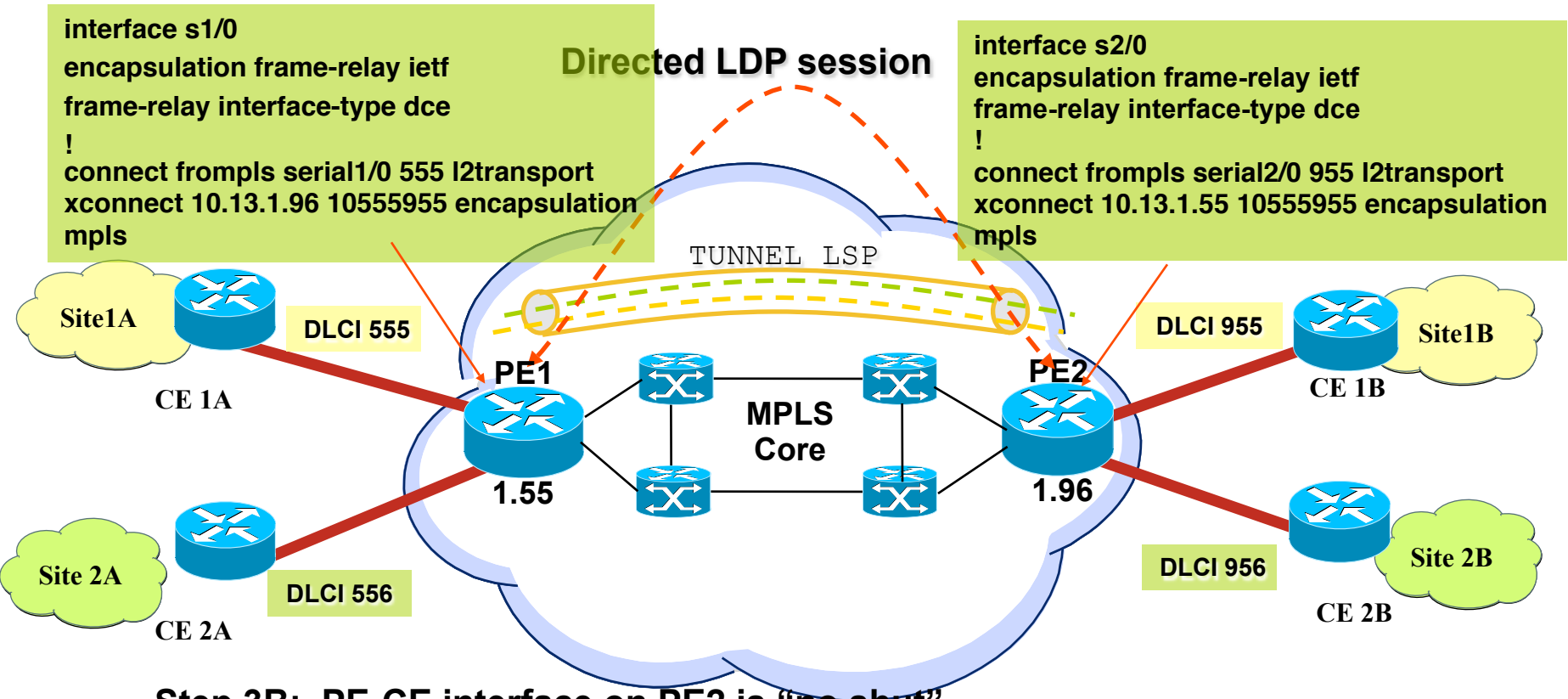
AToM: Control Plane Example



Step 3A: PE-CE interface on PE1 is "no shut"…

1. PE1 will allocate a VC label for DLCI 555
2. binds it to VC ID: 10555955
3. encodes the VC Label TLV with the VC label
4. encodes the VC FEC TLV with the VC ID
5. advertises the label to 10.13.1.96

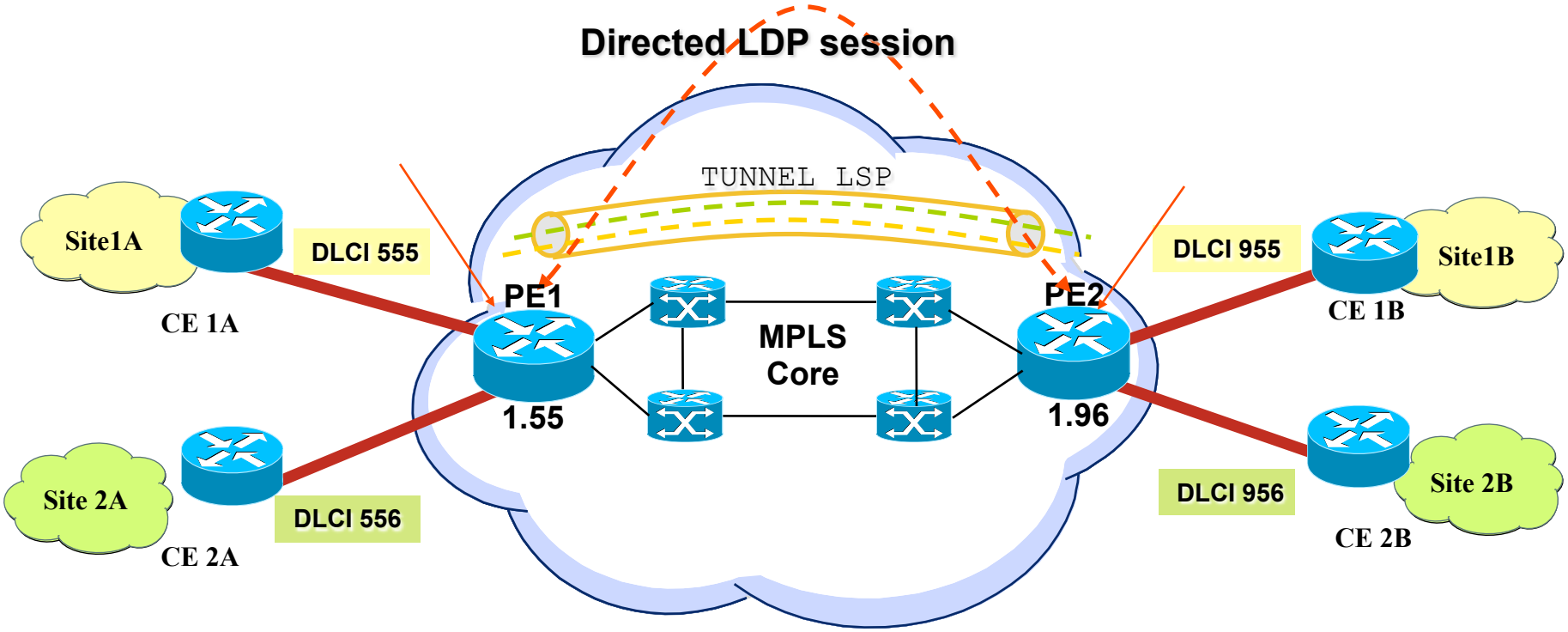
AToM: Control Plane Example



Step 3B: PE-CE interface on PE2 is "no shut"…

1. PE2 will allocate a VC label for DLCI 955
2. binds it to VC ID: 10555955
3. encodes the VC Label TLV with the VC label
4. encodes the VC FEC TLV with the VC ID
5. advertises the label to 10.13.1.55

AToM: Label Withdrawal



PE-CE interface on PE1 is 'shut'…

- PE1 will send a Label Withdrawal message to 10.13.1.96

- status of the VC is down

PE-CE interface on PE2 is 'shut'…

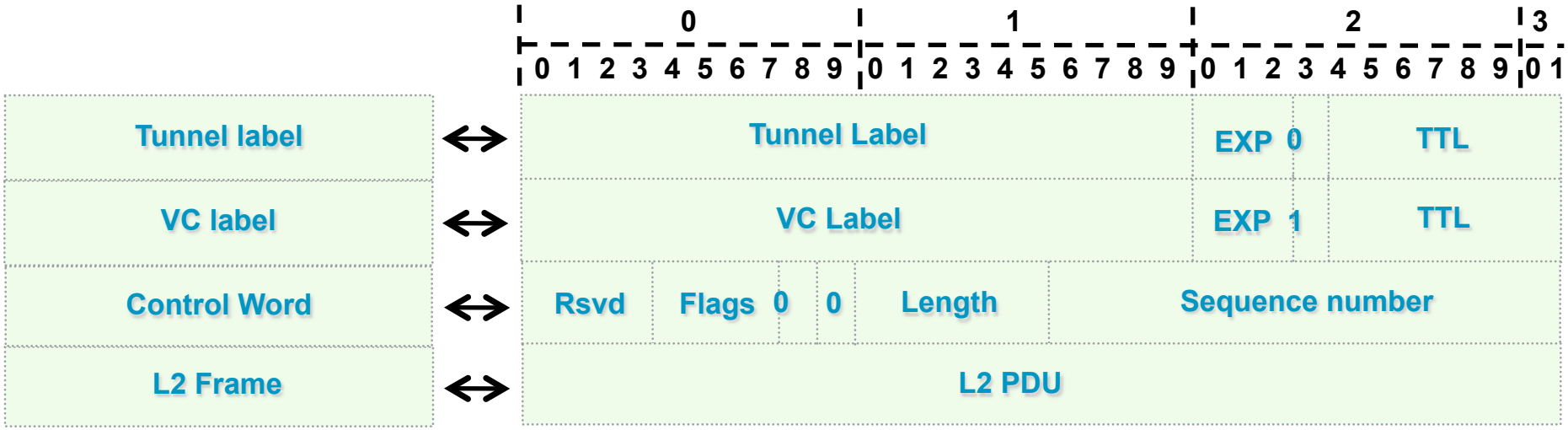
- PE2 will send a Label Withdrawal message to 10.13.1.55

- status of the VC is already down

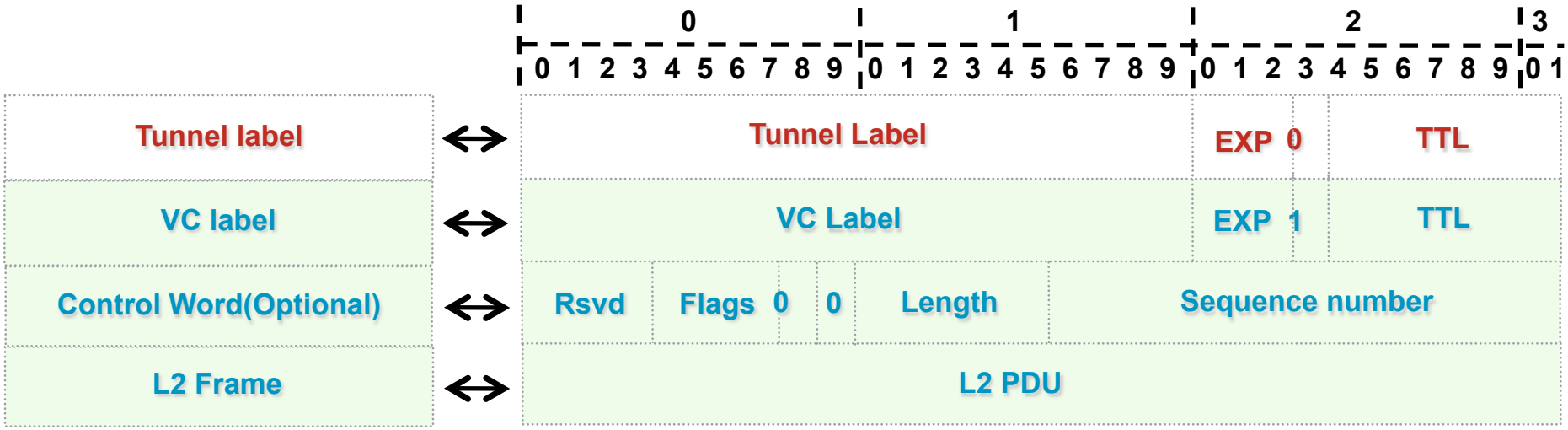
Why LDP signaling is useful between PEs

- To transport circuit status
  - eg. FR: If PE1 sees an issue with dlci 555, it withdraws the VC label so that PE2 can signal the issue on the right via LMI
  - useful for FR, ATM, HDLC, Ethernet…
- In-Sequence delivery
  - Required for ATM and FR. If Ethernet used for non-IP applications, in-sequence delivery is also required
  - PE1 and PE2 can use LDP to synch their sequence numbers after reload/reboot…
- Explicit Goal for PEW3 IETF WG

AToM: Data Plane (Martini Encapsulation)



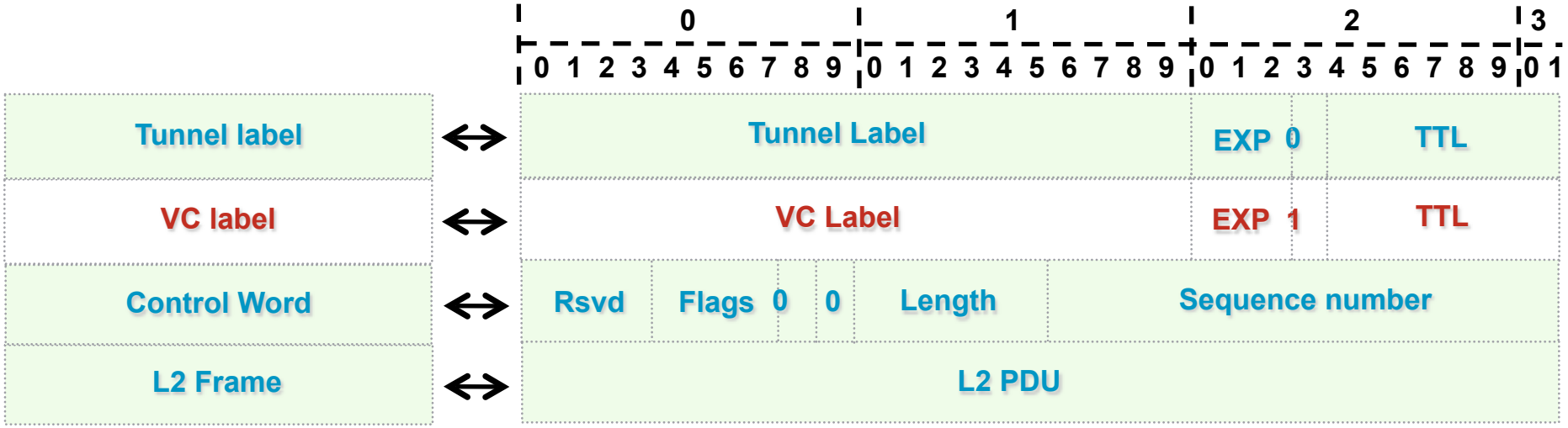
AToM: Data Plane (Martini Encapsulation)
Tunnel Label



Tunnel Label:

- IGP or Outer label that can be distributed by any of the existing mechanisms and is outside the scope of martini draft
- label associated with the tunnel i.e. MPLS LSP or RSVP-TE used to deliver the packet from the ingress PE to egress PE

AToM: Data Plane (Martini Encapsulation) VC Label



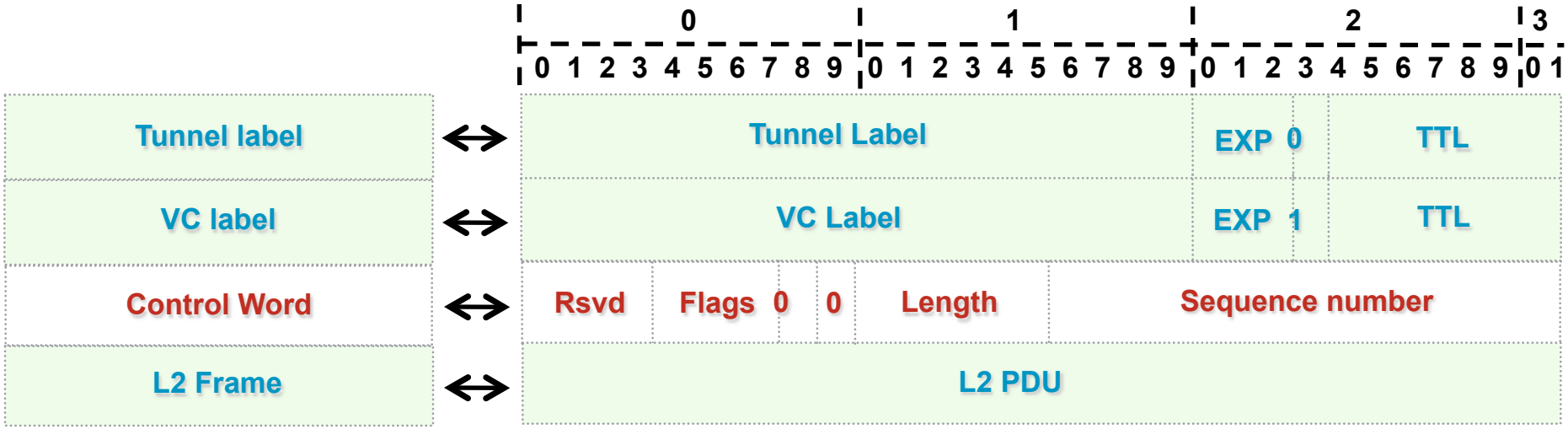
VC Label:

- Inner label that is used by receiving PE to determine the following information and do disposition on the received packet…
  - egress or CE facing interface that the packet should be forwarded to
  - L2 ID such as VLAN or DLCI or PVC used on the CE facing interface

EXP can be set to the values received in the L2 frame, ATM CLP or FR DE bit or it can be set by the PE via CLI or as a result of some QoS policy

TTL is recommended to be set to '2'

AToM: Data Plane (Martini Encapsulation)
Control Word

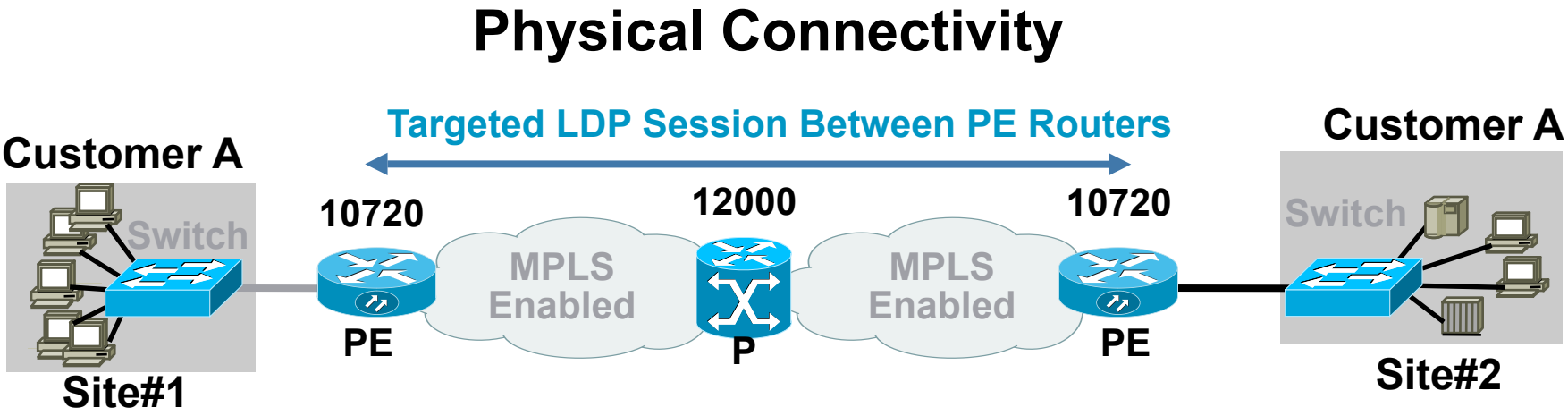


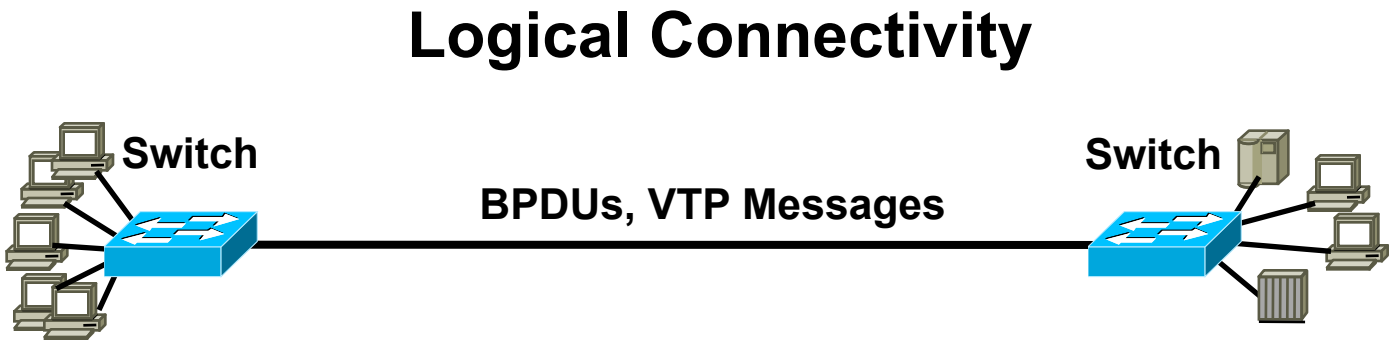
Control Word (CW):

- Optional or Mandatory depending on the type of L2 transport
- Rsvd: Reserved for future use
- Flags: to carry control bits (ATM CLP, FR DE) in the recvd. L2 frame across the MPLS network
- Length: used to indicate the actual packet length if any padding was done to the packet
- Sequence number:
  - provides sequencing capability to detect out of order packets if needed
  - currently not in Cisco's implementation
  - Optional

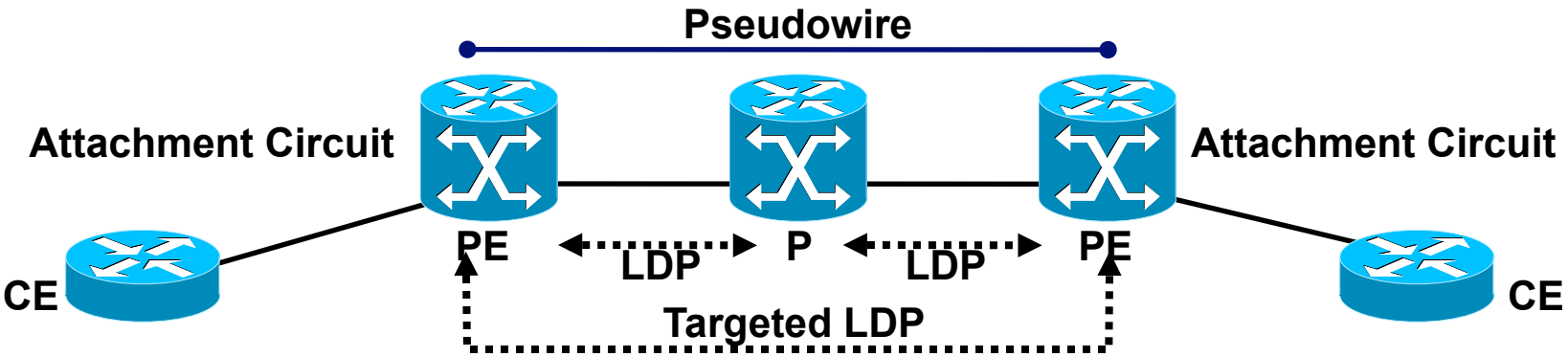
MPLS L2VPNs: EoMPLS

EoMPLS Reference Model



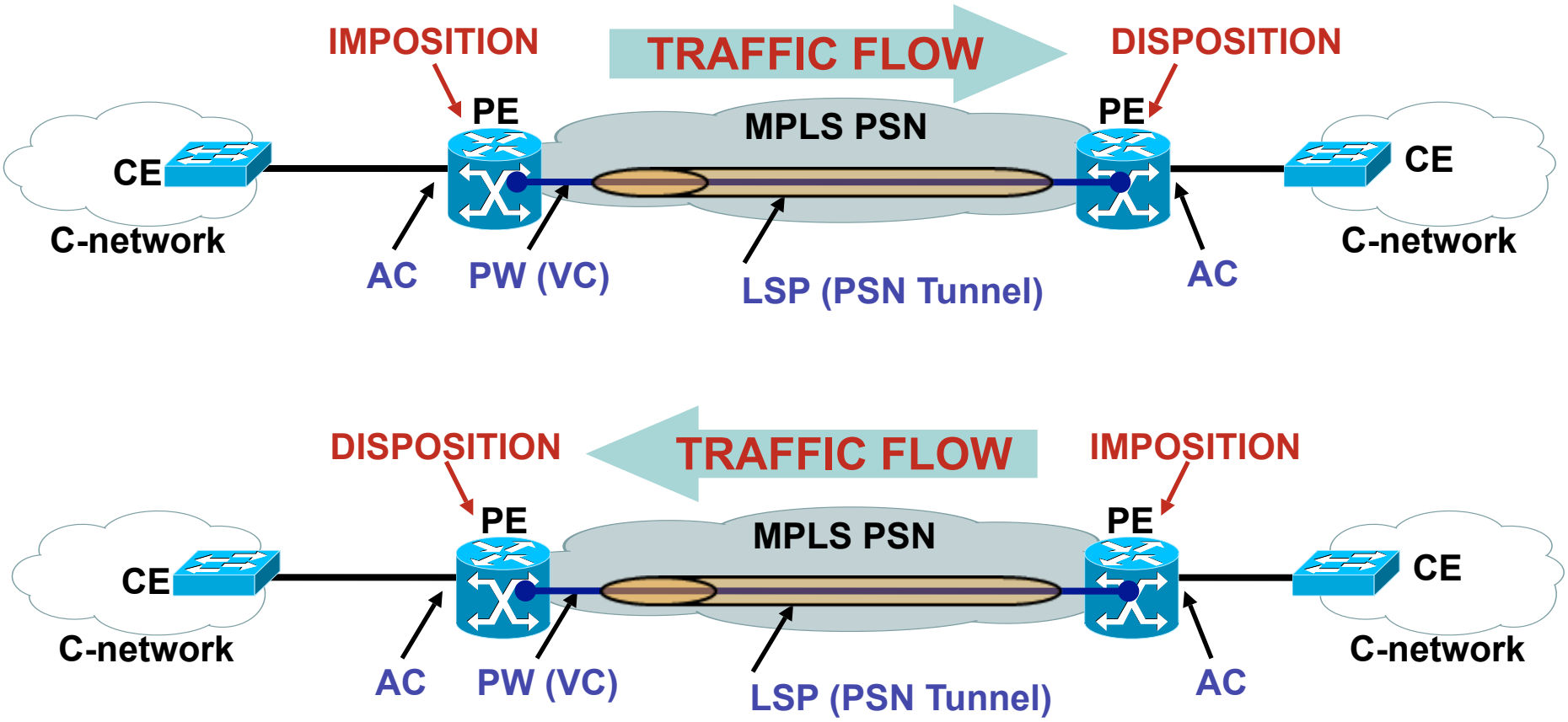


Control Plane



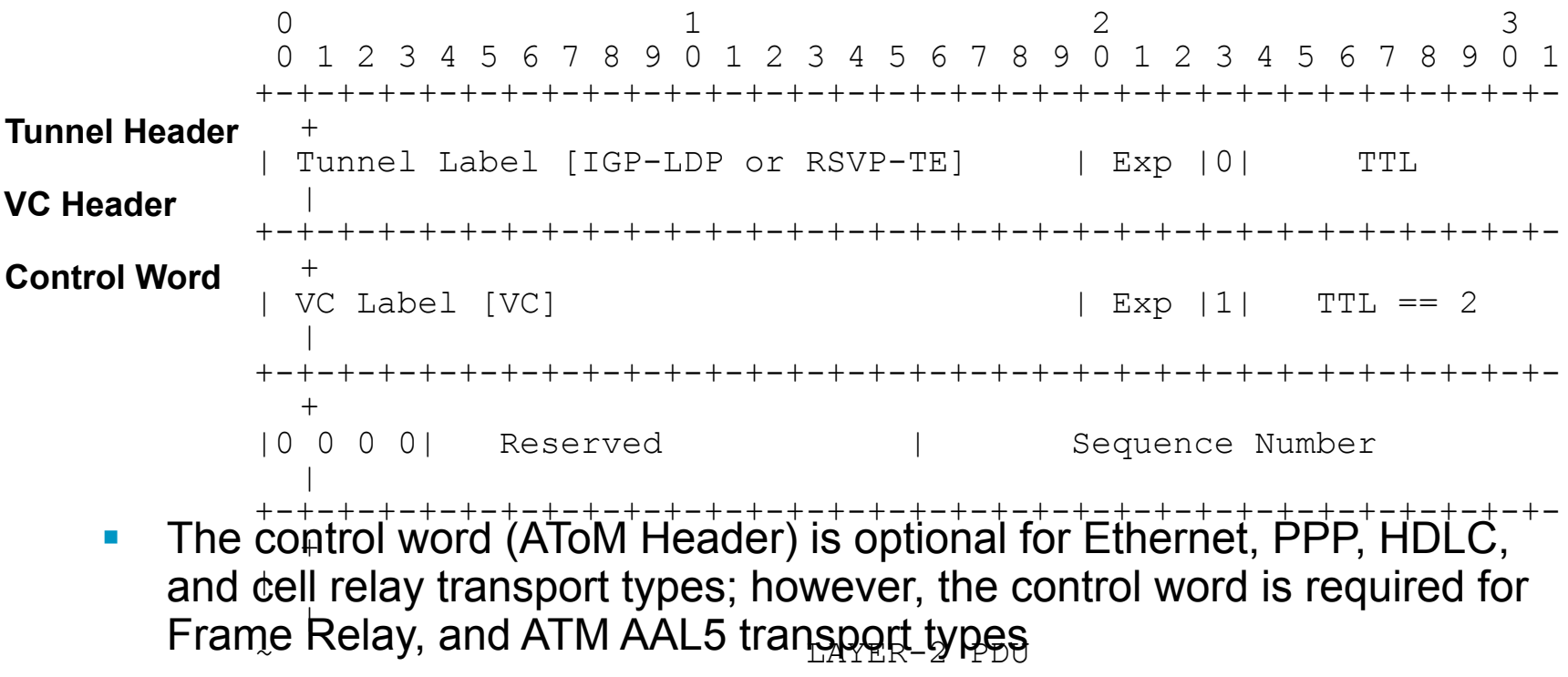
- MPLS in the core
- Targeted (AKA directed) LDP session between PEs
- Targeted LDP session distributes pseudowire (Pw AKA VC) labels
- PE uses per-platform label space (label pool) for both link and targeted LDP sessions (i.e. router_id:0)
- Need LSPs among PEs => Use /32 loopback prefixes

Imposition and Disposition



- AToM needs MPLS LSP from PE to PE
- PEs need to use /32 loopback prefixes for LSPs

Data Plane: EoMPLS Packet Format

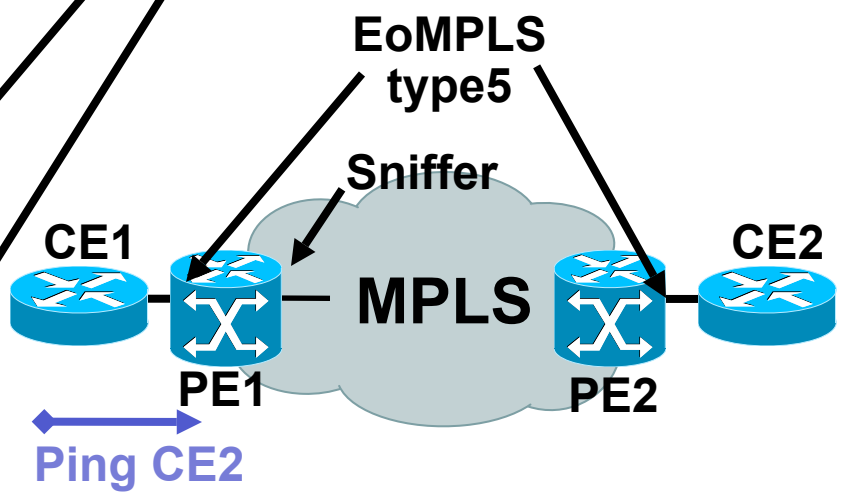


- The control word (AToM Header) is optional for Ethernet, PPP, HDLC, and cell relay transport types; however, the control word is required for Frame Relay, and ATM AAL5 transport types
- First nibble is 0x0 to prevent aliasing with IP Packets over MPLS
- The AToM control word is supported; however, if a peer PE does not support the control word, it is disabled; this negotiation is done by LDP label mapping

Data Plane: A Real Packet

```
Ethernet II
    Destination: xx:xx:xx:xx:xx:xx
    Source: yy:yy:yy:yy:yy:yy
    eType: MPLS Unicast (0x8847)
MultiProtocol Label Switching Header
    MPLS Label: 16
    MPLS Experimental Bits: 0
    MPLS Bottom Of Label Stack: 0
    MPLS TTL: 255
MultiProtocol Label Switching Header
    MPLS Label: 16
    MPLS Experimental Bits: 0
    MPLS Bottom Of Label Stack: 1
    MPLS TTL: 2
AToM EoMPLS Header
    AToM MPLS Control Word: 0x00000000
Ethernet II
    Destination: aa:aa:aa:aa:aa:aa
    Source: bb:bb:bb:bb:bb:bb
    eType: IPv4 (0x0800)
Internet Protocol
    Version: 4
    Header length: 20 bytes
    [snip]
    Time to live: 255
    Protocol: ICMP (0x01)
    Header checksum: 0xa3fd (correct)
    Source: 10.1.2.203 (10.1.2.203)
    Destination: 10.0.0.201 (10.0.0.201)
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
```

Ethernet
Tunnel Shim
VC Shim
Control Word
Ethernet
IPv4
ICMP



AToM Prerequisite Configuration

- Enable [d]CEF globally

  Router(config)#ip cef

- Enable MPLS globally

  Router(config)#mpls ip

- Enable LDP globally as default label distribution protocol

  Router(config)#mpls label protocol ldp

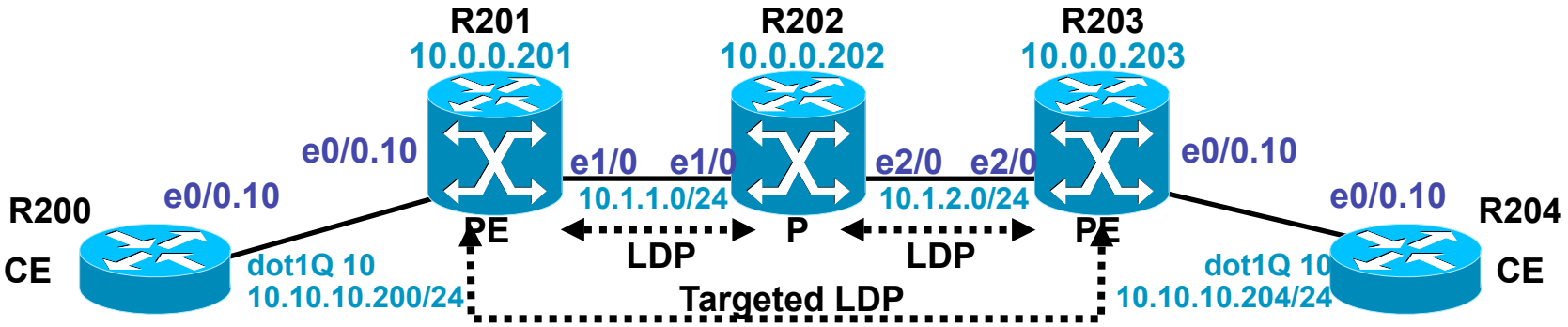
- Specify a loopback interface as LDP Router ID

  Router(config)#mpls ldp router-id loopback <#> [force]

- Enable LDP in the P-PE and P-P interfaces

  Router(config-if)#mpls ip

A Typical Configuration: EoMPLS VLAN




```
hostname R201
!
ip cef
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
!
interface Loopback0
 ip address 10.0.0.201 255.255.255.255
!
interface Ethernet0/0.10
 description *** To R200 ***
 encapsulation dot1Q 10
 no ip directed-broadcast
 no cdp enable
 xconnect 10.0.0.203 10 encapsulation mpls
```

```
hostname R203
!
ip cef
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0 force
!
interface Loopback0
 ip address 10.0.0.203 255.255.255.255
!
pseudowire-class eompls
 encapsulation mpls
!
interface Ethernet0/0.10
 description *** To R204
 encapsulation dot1Q 10
 no ip directed-broadcast
 no cdp enable
 xconnect 10.0.0.201 10 pw-class eompls
```

Verifying the Operation

Working Example

```
R201#show mpls l2transport vc 10 detail
Local interface: Et0/0.10 up, line protocol up, Eth VLAN 10 up
  Destination address: 10.0.0.203, VC ID: 10, VC status: up
    Preferred path: not configured
    Default path: active
    Tunnel label: 17, next hop 10.1.1.202
    Output interface: Et1/0, imposed label stack {17 21}
  Create time: 23:06:37, last status change time: 00:30:47
  Signaling protocol: LDP, peer 10.0.0.203:0 up
    MPLS VC labels: local 19, remote 21
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description: *** To R204 ***
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 1683, send 1777
    byte totals:   receive 565455, send 563328
    packet drops:  receive 0, send 7

R201#
```

VC Status Meaning

- UP—VC can carry data between the 2 endpoints; this means both imposition and disposition are programmed
  - The disposition interfaces is programmed if the VC has been configured and the CE interface is up
  - The imposition interface is programmed if the disposition interface is programmed and we have a remote VC label and an IGP label (Label Switch Path to the peer)
  - The IGP label can be implicit null in a back-to-back configuration
- DOWN—VC is not ready to carry traffic between the two VC endpoints
- ADMINDOWN—VC disabled by a user

EoMPLS: Data Plane Overhead

- At imposition, PE encapsulates CE's Ethernet or VLAN packet to route across MPLS cloud
- These are the associated overheads:

  Transport Header is 6 Bytes DA + 6 Bytes SA + 2 Bytes etype + optional 4 Bytes of VLAN Tag

  There's (at least) 2 levels of MPLS header (Tunnel + VC) each contributing with 4 Bytes

  There is an optional 4-Byte control word

Calculating MTU Requirements for the Core

- Core MTU ≥ Edge MTU + Transport Header + AToM Header (Control Word) + (MPLS Label Stack * MPLS Header Size)
- Edge MTU is the MTU configured in the CE-facing PE's interface
- Examples (all in Bytes):

	Edge	Transport	AToM	MPLS Stack	MPLS Header	Total
EoMPLS Port Mode	1500	14	4 [0]	2	4	1526 [1522]
EoMPLS VLAN Mode	1500	18	4 [0]	2	4	1530 [1526]
EoMPLS Port w/ TE FRR	1500	14	4 [0]	3	4	1530 [1526]

Changing the MTU Size in the Core

- Use the mtu command in PE and P routers to configured at least the calculated minimum MTU

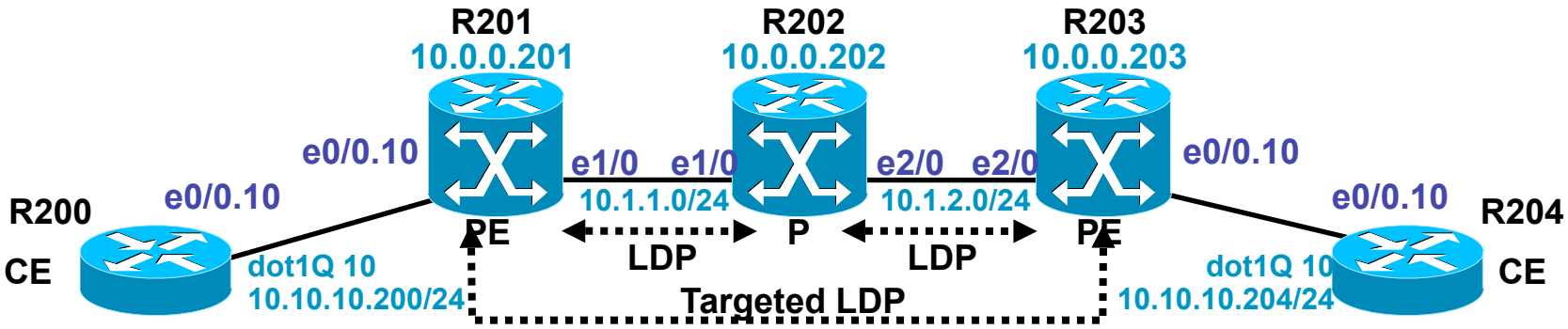
```
Router(config-if)# mtu 1526
```

- Some interfaces (such as FastEthernet interfaces) require the mpls mtu command to change the MTU size

Transport Header Overhead

Transport Type	Transport Header Size
Ethernet Port	14 Bytes
Ethernet VLAN	18 Bytes
Frame Relay DLCI, Cisco Encapsulation	2 Bytes
Frame Relay DLCI, IETF Encapsulation	8 Bytes
HDLC	4 Bytes
PPP	4 Bytes
AAL5	0–32 Bytes

MTU Problem Example




```
R200#ping ip 10.10.10.204 size 1470 timeout 1 df-bit

Type escape sequence to abort.
Sending 5, 1470-byte ICMP Echos to 10.10.10.204, timeout is 1 seconds:
Packet sent with the DF bit set
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/32 ms
R200#
```

```
R200#ping ip 10.10.10.204 size 1471 timeout 1 df-bit

Type escape sequence to abort.
Sending 5, 1470-byte ICMP Echos to 10.10.10.204, timeout is 1 seconds:
Packet sent with the DF bit set
.....
Success rate is 0 percent (0/5)
R200#
```

	Core Overhead
Transport	18 Bytes
AToM	4 Bytes
MPLS Stack	2 Headers
MPLS Header	4 Bytes
Total	30 Bytes

Control Plane: Displaying L2CKT Binding

```
R201#show mpls l2transport binding 10
  Destination Address: 10.0.0.203,  VC ID: 10
    Local Label:  19
        Cbit: 1,    VC Type: Eth VLAN,    GroupID: 0
        MTU: 1500,   Interface Desc: *** To R200 ***
        VCCV Capabilities: Type 1, Type 2
    Remote Label: 21
        Cbit: 1,    VC Type: Eth VLAN,    GroupID: 0
        MTU: 1500,   Interface Desc: *** To R204 ***
        VCCV Capabilities: Type 1, Type 2
R201#
```

Advertised by LDP

Signaled as Interface Parameters

- Cbit: Flags the presence of a control word
- VC Type: (PW type) A 15 bit quantity containing a value which represents the type of PW
- VCCV Capabilities: Virtual Circuit Connection Verification

  Type 1: PWE3 control word (0x0001 as first nibble of CW)

  Type 2: MPLS Router Alert Label
- Interface parameters: used to provide interface specific parameters, such as CE-facing interface MTU

Checking LDP State

```
R201#show mpls ldp discovery
 Local LDP Identifier:
    10.0.0.201:0
    Discovery Sources:
    Interfaces:
        Ethernet1/0 (ldp): xmit/recv
            LDP Id: 10.0.0.202:0
    Targeted Hellos:
        10.0.0.201 -> 10.0.0.203 (ldp): active/passive, xmit/recv
            LDP Id: 10.0.0.203:0
R201#show mpls ldp neighbor
    Peer LDP Ident: 10.0.0.202:0; Local LDP Ident 10.0.0.201:0
        TCP connection: 10.0.0.202.11039 - 10.0.0.201.646
        State: Oper; Msgs sent/rcvd: 54/54; Downstream
        Up time: 00:41:07
        LDP discovery sources:
          Ethernet1/0, Src IP addr: 10.1.1.202
        Addresses bound to peer LDP Ident:
          10.0.0.202      10.1.1.202      10.1.2.202
    Peer LDP Ident: 10.0.0.203:0; Local LDP Ident 10.0.0.201:0
        TCP connection: 10.0.0.203.11010 - 10.0.0.201.646
        State: Oper; Msgs sent/rcvd: 53/53; Downstream
        Up time: 00:38:36
        LDP discovery sources:
          Targeted Hello 10.0.0.201 -> 10.0.0.203, active, passive
        Addresses bound to peer LDP Ident:
          10.0.0.203      10.1.2.203
R201#
```

Verifying Core Forwarding State

- IGP Labels untagged!

```
R201#show mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Untagged    10.1.2.0/24      0          Et1/0       10.1.1.202
17     Untagged    10.0.0.202/32    0          Et1/0       10.1.1.202
18     Untagged    10.0.0.203/32    0          Et1/0       10.1.1.202
19     Untagged    l2ckt(10)        1984544    Et0/0.10    point2point
20     Untagged    l2ckt(20)        45183      Et3/0       point2point
21     Untagged    l2ckt(50)        1435873    Se5/0       point2point
R201#
```

IGP Outgoing Labels Untagged; These Are Label Mappings We Should Have Received from the LDP Neighbor

- The Link LDP session may be down

Core LDP Problems

- Indeed, IGP LDP session is down

```
R201#show mpls ldp neighbor
    Peer LDP Ident: 10.0.0.203:0; Local LDP Ident 10.0.0.201:0
        TCP connection: 10.0.0.203.11027 - 10.0.0.201.646
        State: Oper; Msgs sent/rcvd: 12/12; Downstream
        Up time: 00:01:30
        LDP discovery sources:
          Targeted Hello 10.0.0.201 -> 10.0.0.203, active, passive
        Addresses bound to peer LDP Ident:
          10.0.0.203      10.1.2.203
R201#
R201#show mpls ldp discovery
 Local LDP Identifier:
    10.0.0.201:0
    Discovery Sources:
    Interfaces:
        Ethernet1/0 (ldp): xmit
    Targeted Hellos:
        10.0.0.201 -> 10.0.0.203 (ldp): active/passive, xmit/recv
            LDP Id: 10.0.0.203:0
R201#
```

Only Targeted Session Is UP

Transmitting LDP Hellos but Not Receiving Any; Good State Is xmit/recv

Note of Caution

- The operational status in `show mpls interfaces' is not an indicator of neighbor UP

```
R201#show mpls interfaces
Interface              IP             Tunnel   Operational
Ethernet1/0            Yes (ldp)      No       Yes
R201#
```

- The operational status merely shows that some application has enabled processing of MPLS packets
- In fact, in Inter-AS scenarios the interface to which the MP-eBGP VPNv4 neighbor is connected will show operational Yes even without `mpls ip' configured on the interface

Use IP/ICMP-Based Traceroute from PE

- Traceroute with core LDP down (but targeted LDP UP)

```
R201#traceroute 10.0.0.203

Type escape sequence to abort.
Tracing the route to 10.0.0.203

  1 10.1.1.202 28 msec 24 msec 40 msec
  2 10.1.2.203 32 msec 44 msec 36 msec
R201#
```

- Traceroute with core LDP UP

```
R201#traceroute 10.0.0.203


Type escape sequence to abort.
Tracing the route to 10.0.0.203


  1 10.1.1.202 [MPLS: Label 17 Exp 0] 36 msec 20 msec 40 msec
  2 10.1.2.203 28 msec 40 msec 28 msec
R201#
```

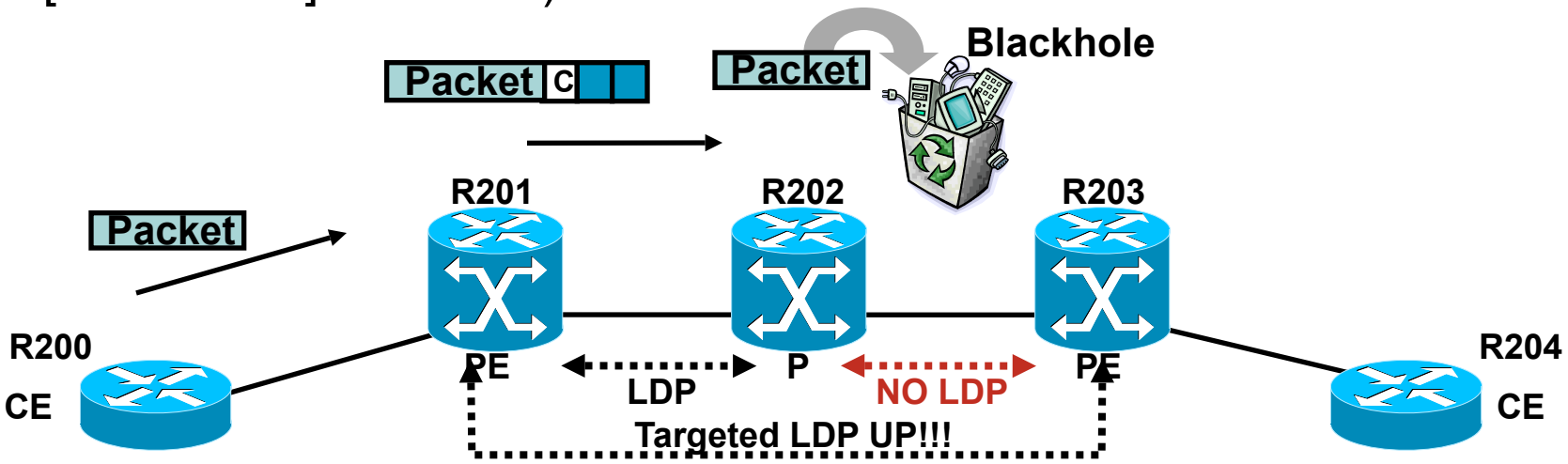
ICMP Extensions for MultiProtocol Label Switching

No MPLS Shim Due to PHP, Would Be 0 If `mpls ldp Explicit-null'

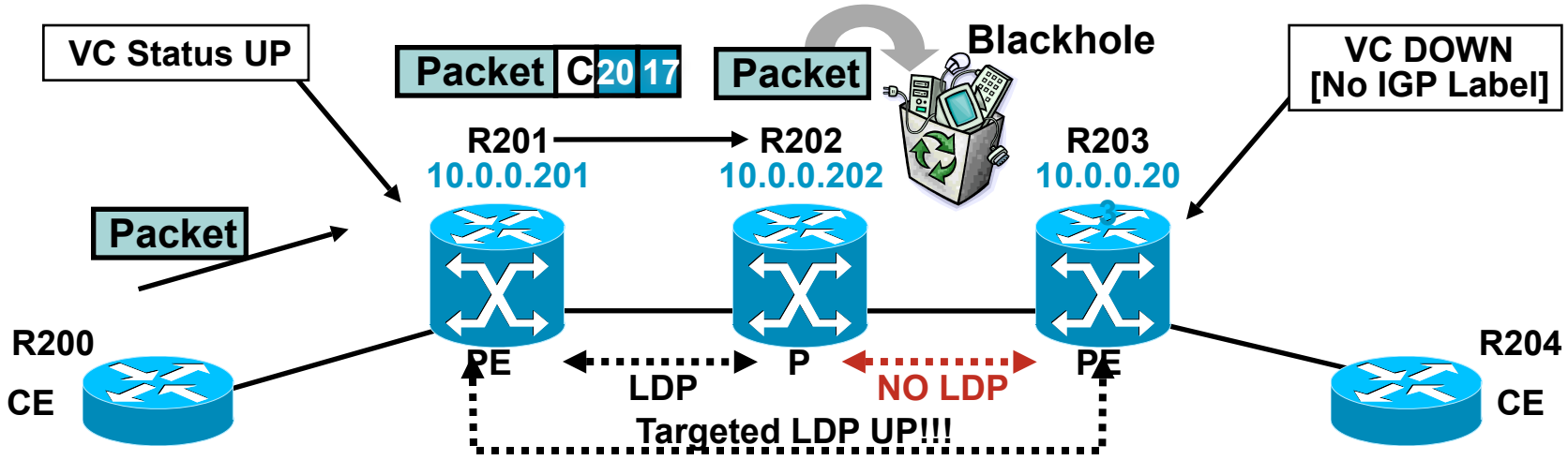
- Remember that PE receives Layer 2 (not IP) packet from CE; here there's no concept of `mpls ip propagate-ttl'

Core LDP Important Note

- The targeted LDP session can be UP even if the Link LDP session (core LDP) is down; all VCs would be UP
- Hey, it just needs a TCP connection that can run over TCPoIP as opposed to TCPoIPoMPLS
- However, the data path is broken (need end-to-end LSP; no IP PID [Protocol ID] for MPLS)



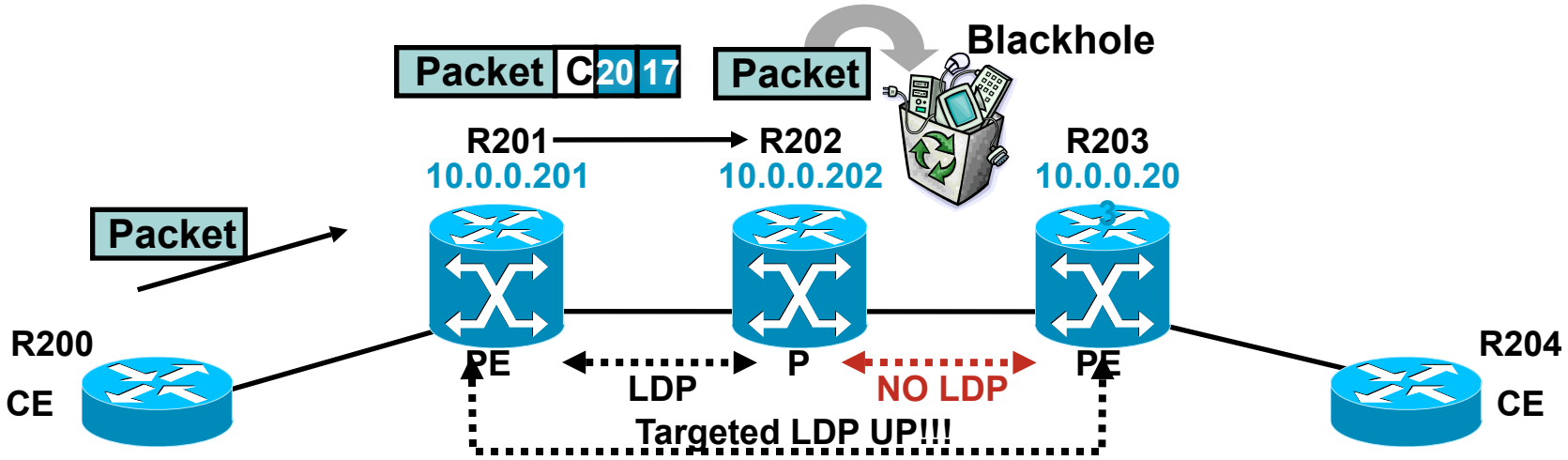
Core Network Dataplane Blackhole




```
R201#show mpls l2transport vc 10 detail  | i status:|Out
   Destination address: 10.0.0.203, VC ID: 10, VC status: up
     Output interface: Et1/0, imposed label stack {17 20}
R201#show mpls forwarding-table 10.0.0.203
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
22     17          10.0.0.203/32    0          Et1/0       10.1.1.202
R201#
```

```
R202#show mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     0           10.0.0.201/32    0          Et1/0       10.1.1.201
17     Untagged    10.0.0.203/32    44773      Et2/0       10.1.2.203
R202#
```

Core Network Dataplane Blackhole




```
R201#traceroute 10.0.0.203

Type escape sequence to abort.
Tracing the route to 10.0.0.203

  1 10.1.1.202 [MPLS: Label 17 Exp 0] 16 msec 44 msec 20 msec
  2 10.1.2.203 28 msec 28 msec 48 mse
R201#
```

With ICMP/IP Based Traceroute, We Cannot Tell If the Absence of MPLS Header Is Because:
1. PHP :-)
2. UNTAGGED !!! :-(

- But data path is broken…
- How do we troubleshoot this one?

With MPLS Embedded Management

- Virtual Circuit Connection Verification (VCCV) is smart enough to differentiate POP vs. untagged
  - POP—the next hop advertised an implicit NULL label for the destination and that this router pops the top label
  - Untagged—there is no label for the destination from the next hop; remove all labels in the stack
- “Detecting MPLS dataplane failures” has PING and traceroute modes
- MPLS echo request and echo reply are UDP packets to port 3503 using label stack to be switched inband the LSP

Virtual Circuit Connection Verification

- MPLS ping has `IPv4', `pseudowire' and `traffic-eng' modes
- MPLS traceroute has `IPv4' and `traffic-eng' modes
- VCCV consists of:

  Signaling component in LDP label mapping for VC FEC with VCCV interface parameter

  Switching component so that VCCV "control" packet is treated as AToM payload from switching standpoint

  VCCV disposition capabilities are:

  Type 1—Uses PID in AToM control word (0x1) [default]

  Type 2—Uses MPLS router alert label

MPLS Pseudowire PING

```
R201#ping 10.0.0.203


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.203, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/27/36 ms
R201#
```

```
R201#ping mpls pseudowire 10.0.0.203 10
Sending 5, 100-byte MPLS Echos to 10.0.0.203/0,
      timeout is 2 seconds, send interval is 0 msec:


Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target


Type escape sequence to abort.
.....
Success rate is 0 percent (0/5)
R201#
```

Virtual Circuit Connection Verification

- Two reply modes:

  IPv4—replies with UDP/IPv4 packet; do not have control over whether the reply is forwarded as IP or IPoMPLS; if no reply, use the next option

  Router alert—adds the router alert option to the IP header (RFC2113); this means that Cisco routers will processed switched (by the RP) at each intermediate hop

- The `router-alert' option bypasses hardware and linecard forwarding table inconsistencies by being punted to the process switching path, but it's more expensive

MPLS Traceroute for IPv4

```
R201#traceroute mpls ipv4 10.0.0.203/32
Tracing MPLS Label Switched Path to 10.0.0.203/32, timeout is 2 seconds



Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target


Type escape sequence to abort.
  0 10.1.1.201 MRU 1500 [Labels: 17 Exp: 0] ! MRU in show mpls forw det
R 1 10.1.2.202 MRU 1504 [No Label] 60 ms                ! Untagged
! 2 10.1.2.203 80 ms
R201#
```

```
R201#traceroute mpls ipv4 10.0.0.203/32
Tracing MPLS Label Switched Path to 10.0.0.203/32, timeout is 2 seconds



Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target


Type escape sequence to abort.
  0 10.1.1.201 MRU 1500 [Labels: 17 Exp: 0]
R 1 10.1.2.202 MRU 1500 [Labels: 0 Exp: 0] 40 ms  ! If Explicit NULL
! 2 10.1.2.203 80 ms
```

MPLS Traceroute for IPv4

```
R201#traceroute mpls ipv4 10.0.0.203/32
Tracing MPLS Label Switched Path to 10.0.0.203/32, timeout is 2 seconds



Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target


Type escape sequence to abort.
  0 10.1.1.201 MRU 1500 [Labels: 17 Exp: 0]
R 1 10.1.2.202 MRU 1504 [implicit-null] 64 ms  ! Implicit NULL
! 2 10.1.2.203 92 ms
R201#
```

```
R201#traceroute mpls ipv4 10.0.0.203/32
Tracing MPLS Label Switched Path to 10.0.0.203/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target

Type escape sequence to abort.
  0 10.1.1.201 MRU 1500 [Labels: 17 Exp: 0]
R 1 10.1.2.202 MRU 1504 [No Label] 76 ms
. 2 *
[snip]
. 6 *
R201#
```

Verifying Forwarding State for L2CKTs

- No local labels for PW; targeted LDP down?

```
R201#show mpls forwarding-table | include Prefix|l2ckt
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
R201#
```

- Local labels assigned to PW (VC)

```
R201#show mpls forwarding-table | include Prefix|l2ckt
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
19     Untagged    l2ckt(10)        1952531    Et0/0.10   point2point
20     Untagged    l2ckt(20)        3024       Et3/0      point2point
21     Untagged    l2ckt(50)        1404829    Se5/0      point2point
R201#
```

Labels Advertised

VC ID (Pw ID)

P2p Adjacency

EoMPLS VC Status DOWN…

- …but everything seems fine on the PEs
- VC Type (PW Type) mismatch

  PW Type—A 15 bit quantity containing a value which represents the type of PW

  VC type—0x0004 is used for IEEE 802.1Q VLAN over MPLS application; ISL is not supported

  VC type—0x0005 is used for Ethernet port tunneling application (port transparency)

- MTU mismatch

  MTU is carried as interface parameter in label mapping message for VC (PW) FEC

VC Type Mismatch

- Show AToM bindings

```
R201#show mpls l2transport binding 10
  Destination Address: 10.0.0.203,  VC ID: 10
    Local Label:  19
        Cbit: 1,     VC Type: Ethernet,     GroupID: 0
        MTU: 1500,    Interface Desc: n/a
        VCCV Capabilities: Type 1, Type 2
    Remote Label: 21
        Cbit: 1,     VC Type: Eth VLAN,     GroupID: 0
        MTU: 1500,    Interface Desc: *** To R204 ***
        VCCV Capabilities: Type 1, Type 2
R201#
```

VC Type Mismatch !!!

- AToM VC is not enough to find the problem

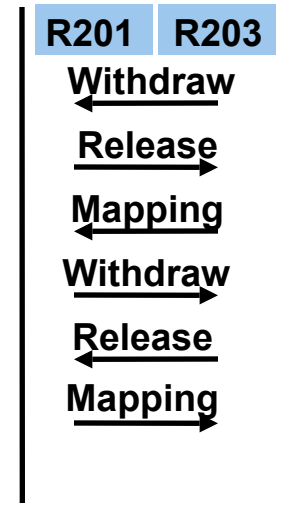
```
R201#show mpls l2transport vc  10 detail
Local interface: Et3/0 up, line protocol up, Ethernet up
  Destination address: 10.0.0.203, VC ID: 10, VC status: down
    Tunnel label: not ready
    Output interface: unknown, imposed label stack {}
  Create time: 00:47:45, last status change time: never
  Signaling protocol: LDP, peer 10.0.0.203:0 up
    MPLS VC labels: local 19, remote 21
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description: *** To R204 ***
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 0, send 0
    byte totals:   receive 0, send 0
    packet drops:  receive 0, send 0
```

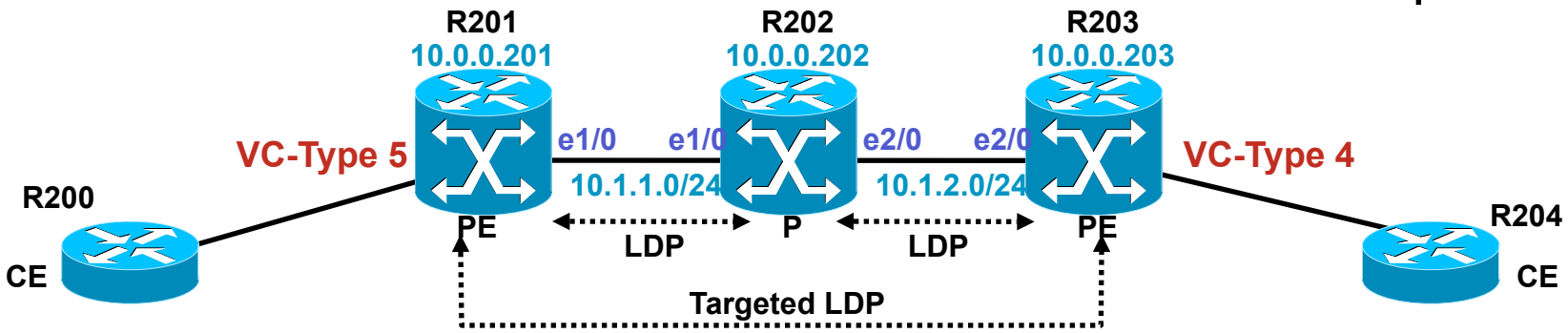
VC Down
But All Seems Fine

VC Type Mismatch

- Enable `debug mpls l2transport signaling message'
- Need to shut/no shut to trigger sending new LDP msgs

```
R201#
*Apr 21 19:17:44.308: AToM LDP [10.0.0.203]: Received label withdraw msg, id 3964
vc type 4, cbit 1, vc id 10, group id 0, vc label 22, status 0, mtu 0
*Apr 21 19:17:44.400: AToM LDP [10.0.0.203]: Sending label release msg
vc type 4, cbit 1, vc id 10, group id 0, vc label 22, status 0, mtu 0
*Apr 21 19:17:58.188: AToM LDP [10.0.0.203]: Received label mapping msg, id 3965
vc type 4, cbit 1, vc id 10, group id 0, vc label 21, status 0, mtu 1500
*Apr 21 19:18:42.100: AToM LDP [10.0.0.203]: Sending label withdraw msg
vc type 5, cbit 1, vc id 10, group id 0, vc label 19, status 0, mtu 1500
*Apr 21 19:18:42.512: AToM LDP [10.0.0.203]: Received label release msg, id 3968
vc type 5, cbit 1, vc id 10, group id 0, vc label 19, status 0, mtu 0
*Apr 21 19:18:43.500: AToM LDP [10.0.0.203]: Sending label mapping msg
vc type 5, cbit 1, vc id 10, group id 0, vc label 20, status 0, mtu 1500
R201#
```





VC Type Mismatch

- debug mpls l2transport vc event

```
R203# ! Remote shut
1d04h: AToM MGR [10.0.0.201, 10]: Delete remote vc label binding
R203# ! Remote no shut
1d04h: AToM MGR [10.0.0.201, 10]: Remote end up
1d04h: AToM MGR [10.0.0.201, 10]: Mismatch vc type in remote label binding, local 4, remote 5
R203#
```

Remote PE

VC Id

- debug mpls l2transport vc fsm

```
R203# ! Remote shut
1d04h: AToM MGR [10.0.0.201, 10]: Event remote down, state changed from establishing to local ready
R203# ! Remote no shut
1d04h: AToM MGR [10.0.0.201, 10]: Event remote up, state changed from local ready to establishing
1d04h: AToM MGR [10.0.0.201, 10]: Event remote invalidated, state changed from establishing to establishing
1d04h: AToM MGR [10.0.0.201, 10]: Take no action
R203#
```

MTU Mismatch

- debug mpls l2transport vc event

```
R203#
1d05h: AToM MGR [10.0.0.201, 50]: Remote end up
1d05h: AToM MGR [10.0.0.201, 50]: Mismatch MTU in remote label binding, local 1500, remote 4400
R203#
```

- debug mpls l2transport vc fsm

```
R203#
1d05h: AToM MGR [10.0.0.201, 50]: Event remote up, state changed from local ready to establishing
1d05h: AToM MGR [10.0.0.201, 50]: Event remote invalidated, state changed from establishing to establishing
1d05h: AToM MGR [10.0.0.201, 50]: Take no action
R203#
```

- Note: Changing `MTU' will cause a new LDP Mapping message to be sent for the VC (PW) FEC
- Changing the `description' will not, and shut/no shut is needed to generate the signaling message

A Successful Pseudowire Establishment

- debug mpls l2transport vc event

```
R203#
1d05h: AToM MGR [10.0.0.201, 50]: Remote end up
1d05h: AToM MGR [10.0.0.201, 50]: Validate vc, activating data plane
1d05h: AToM SMGR: Submit Imposition Update
1d05h: AToM SMGR: Submit Disposition Update
1d05h: AToM SMGR: Submit SSM event
1d05h: AToM SMGR [10.0.0.201, 50]: Event Imposition Enable, imp-ctrlflag 83, remote vc label 20
1d05h: AToM SMGR [10.0.0.201, 50]: Imposition Programmed, Output Interface: Et1/0
1d05h: AToM SMGR [10.0.0.201, 50]: State [Provisioned->Imposition Rdy]
1d05h: AToM SMGR [10.0.0.201, 50]: Event Disposition Enable, disp-ctrlflag 3, local vc label 16
1d05h: AToM SMGR [10.0.0.201, 50]: State [Imposition Rdy->Imposition/Disposition Rdy]
1d05h: AToM SMGR: Event SSM event
1d05h: AToM SMGR [10.0.0.201, 50]: sucessfully processed ssm provision request pwid 300001F
1d05h: AToM SMGR [10.0.0.201, 50]: Send COMPLETE signal to SSM
1d05h: AToM SMGR [10.0.0.201, 50]: sucessfully setup sss switch for pwid 300001F
1d05h: AToM SMGR: Submit SSM event
1d05h: AToM SMGR: Event SSM event
1d05h: AToM SMGR [10.0.0.201, 50]: sucessfully processed ssm bind for  pwid 300001F
1d05h: AToM MGR [10.0.0.201, 50]: Receive SSM dataplane up notification
1d05h: AToM MGR [10.0.0.201, 50]: Dataplane activated
R203#
```

Pseudowire Dataplane Is Up Only if Imposition/Disposition Are Successfully Programmed

What If the Core Uses Traffic Engineering?

- Need to use the command `preferred-path {interface | peer}' under the `pseudowire-class'; have in mind that:
  - The selected path must be a label switched path (LSP) destined to the peer PE router
  - If you specify a tunnel (selecting interface):
    - The tunnel must be an MPLS traffic engineering tunnel
    - The tunnel tailend must be on the remote PE router
  - If you specify an IP address (selecting peer):
    - The address must be the IP address of a *loopback* interface on the *remote PE router*, not necessarily the LDP router-id address; peer means *targeted LDP peer*
    - The address must have a /32 mask
    - There must be an LSP destined to that selected address
    - The LSP does not have to be a TE tunnel

Tunnel Selection Details

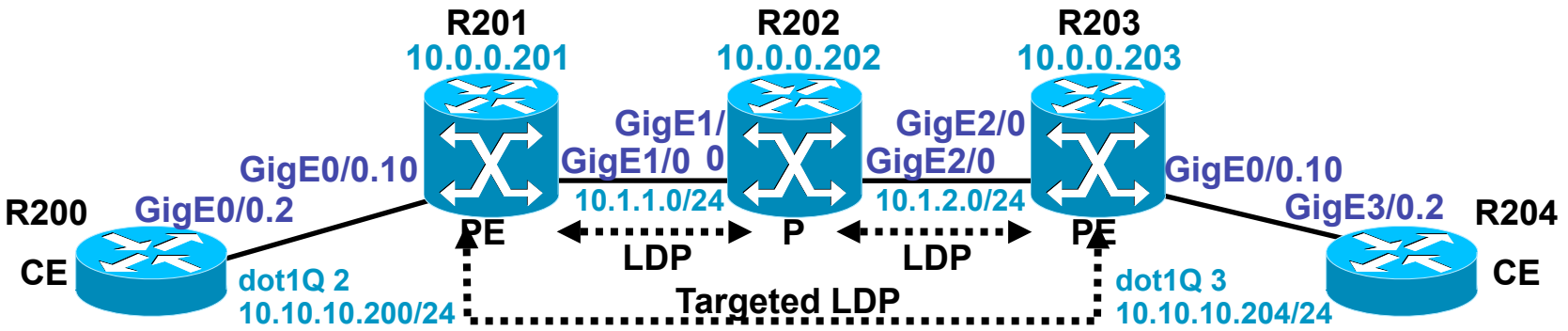
- Tunnel selection configuration

```
pseudowire-class AToM_FRoMPLS
 encapsulation mpls
 preferred-path interface Tunnel1
!
interface Tunnel1
 description AToM_Tunnel
 ip unnumbered Loopback1
 mpls ip
 tunnel destination 192.168.200.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 dynamic
connect VPN200 POS3/0 200 l2transport
 xconnect 192.168.200.2 200 encapsulation mpls pw-class AToM_FRoMPLS
```

- Tunnel selection verification

```
PE1#show mpls l2transport vc 200 de
Local interface: PO3/0 up, line protocol up, FR DLCI 200 up
  Destination address: 192.168.200.2, VC ID: 200, VC status: up
    Data plane status: imposition OK, disposition OK
    Output interface: Tu1, imposed label stack {24 35}
    Preferred path: Tunnel1,  active
    Default path: not supported
  Create time: 00:28:33, last status change time: 00:27:09
  Signaling protocol: LDP, peer 192.168.200.2:0 up
    MPLS VC labels: local 39, remote 24
    Group ID: local 0, remote 0
    MTU: local 4470, remote 4470
    Remote interface description: to CE2
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 30, send 30
    byte totals:   receive 11295, send 11745
    packet drops:  receive 0, send 0
```

VLAN ID Rewrite Configuration



R201

```
interface GigabitEthernet0/0.2
  encapsulation dot1Q 2
  no ip directed-broadcast
  no cdp enable
  xconnect 10.0.0.203 2 encapsulation mpls
    remote circuit id 3
```

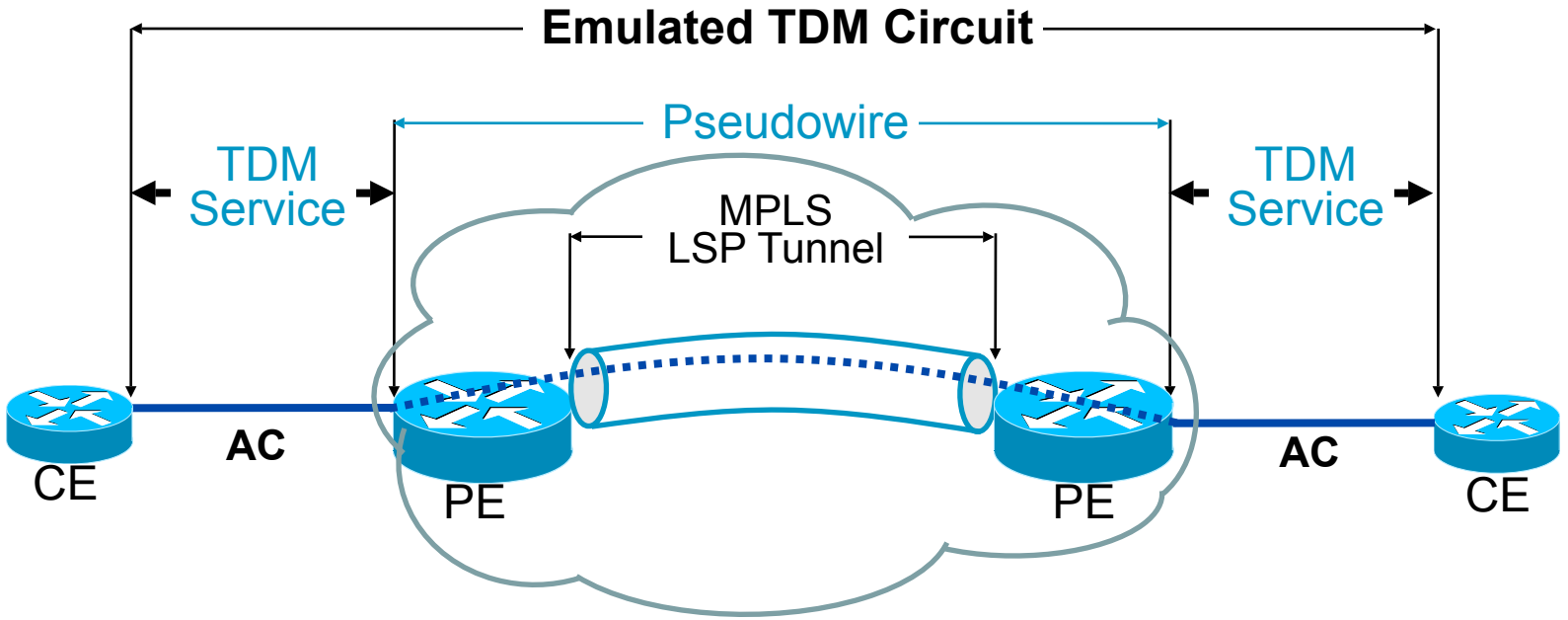
XConnect submode

R203

```
interface GigabitEthernet3/0.2
  encapsulation dot1Q 3
  no ip directed-broadcast
  no cdp enable
  xconnect 10.0.0.201 2 encapsulation mpls
    remote circuit id 2
```

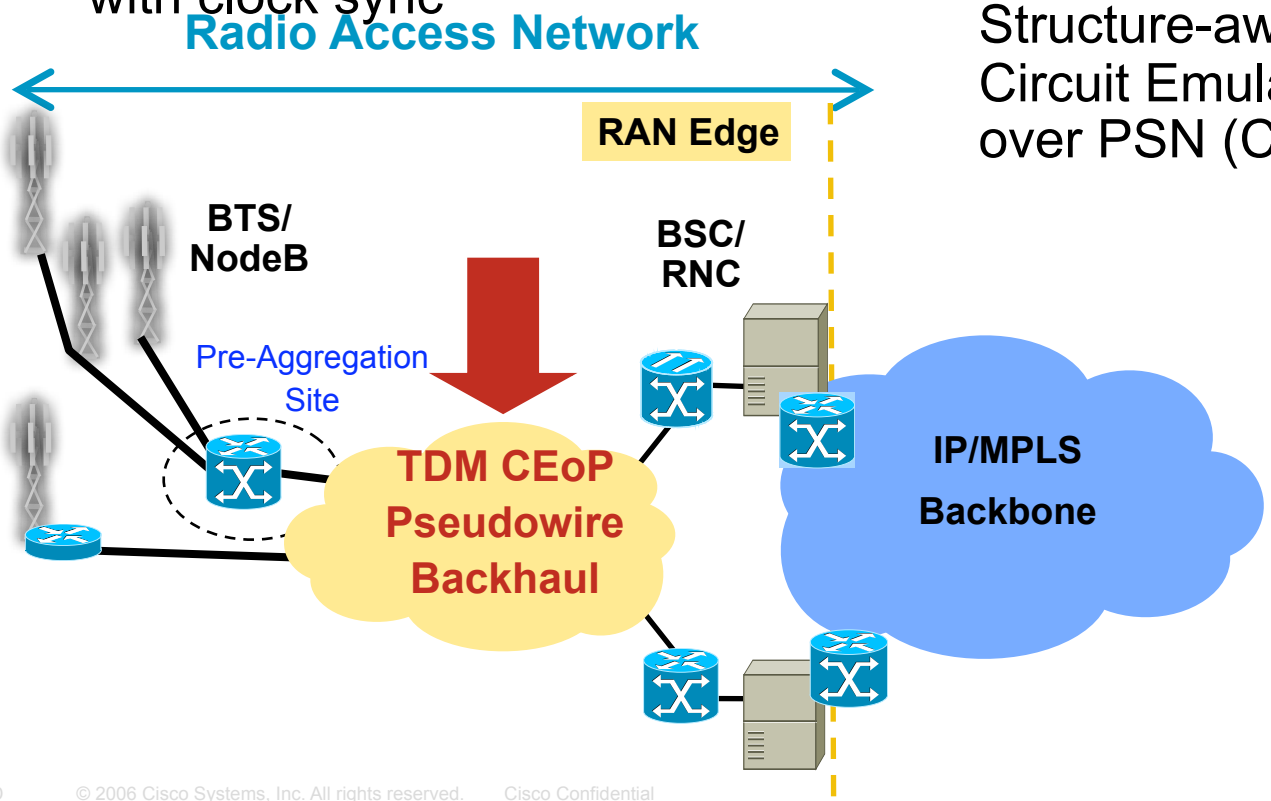
Circuit Emulation over Packet (CEoP)

- Circuit Emulation over Packet (CEoP) allows customers to provide TDM circuit service over a Packet Switched Network (PSN)
  - CEoP emulates T1/E1, T3/E3 and OC3/STM-1, unstructured and structured, down to nxDS0 circuits
  - Available for AToM (MPLS) now; L2TPv3 (IP) in future



Circuit Emulation over Packet (CEoP) in the Radio Access Network (RAN)

- Synchronous point-to-point transport
  - Supports TDM and SONET/SDH with clock sync
- Structured and Unstructured framing
  - Structure-Agnostic TDM over Packet (SAToP)
  - Structure-aware TDM Circuit Emulation Service over PSN (CESoPSN)



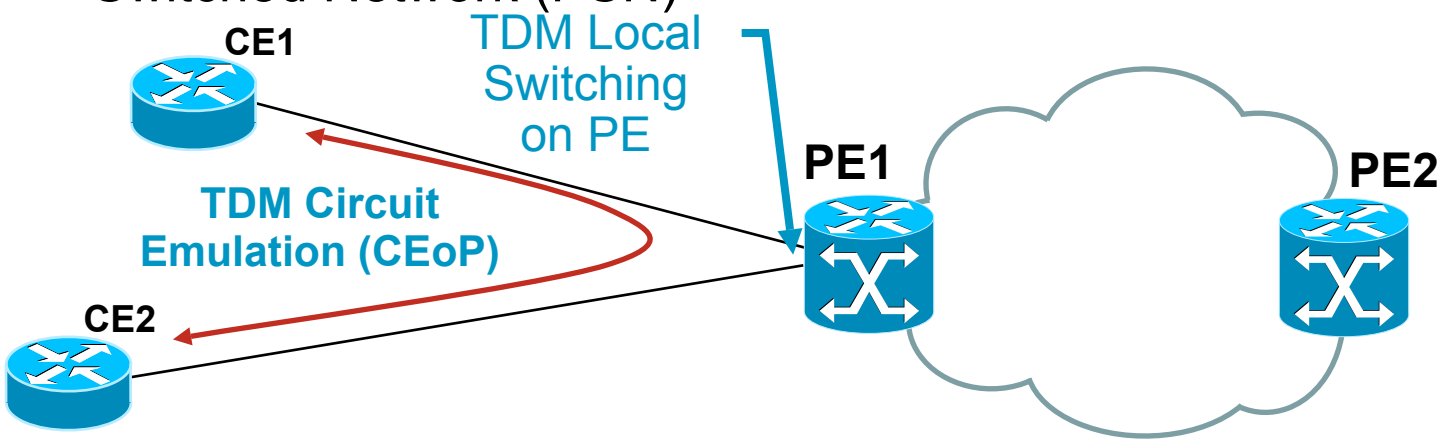
TDM Local Switching

- TDM Local Switching

  Enables connection of TDM circuits among Customer Edge (CE) devices attached to the same Provider Edge router (PE)

- Circuit Emulation over Packet (CEoP)

  Allows customers to provide TDM circuit service over a Packet Switched Network (PSN)



Available on Cisco 7600 series routers with CEoP SPA and SIP-400