

# Deploying MPLS L3VPN

Nurul Islam Roman ([nurul@apnic.net](mailto:nurul@apnic.net))

# Abstract

- **This session** describes the implementation of **IP Virtual Private Networks (IP VPNs)** using **MPLS**. It is the most common Layer 3 VPN technology, as standardized by IETF RFC2547/4364, realizing IP connectivity between VPN site and MPLS network.
- Service Providers have been using IP VPN to provide scalable site-to-site/WAN connectivity to Enterprises/SMBs' for more than a decade. Enterprises have been using it to address **network segmentation (virtualization)** and traffic separation) inside the site e.g. Campus, Data Center. This technology realizes IP connectivity between VPN site and MPLS network.
- The session will cover:
  - IP VPN Technology Overview (RFC2547/RFC4364)
  - IP VPN Configuration Overview
  - IP VPN-based services (multihoming, Hub&Spoke, extranet, Internet, NAT, VRF-lite, etc.)
  - Best Practices

# Terminology

- LSR: label switch router
- LSP: label switched path
  - The chain of labels that are swapped at each hop to get from one LSR to another
- VRF: VPN routing and forwarding
  - Mechanism in Cisco IOS® used to build per-customer RIB and FIB
- MP-BGP: multiprotocol BGP
- PE: provider edge router interfaces with CE routers
- P: provider (core) router, without knowledge of VPN
- VPNv4: address family used in BGP to carry MPLS-VPN routes
- RD: route distinguisher
  - Distinguish same network/mask prefix in different VRFs
- RT: route target
  - Extended community attribute used to control import and export policies of VPN routes
- LFIB: label forwarding information base
- FIB: forwarding information base

# Agenda

- IP/VPN Overview
- IP/VPN Services
- Best Practices
- Conclusion

# Agenda

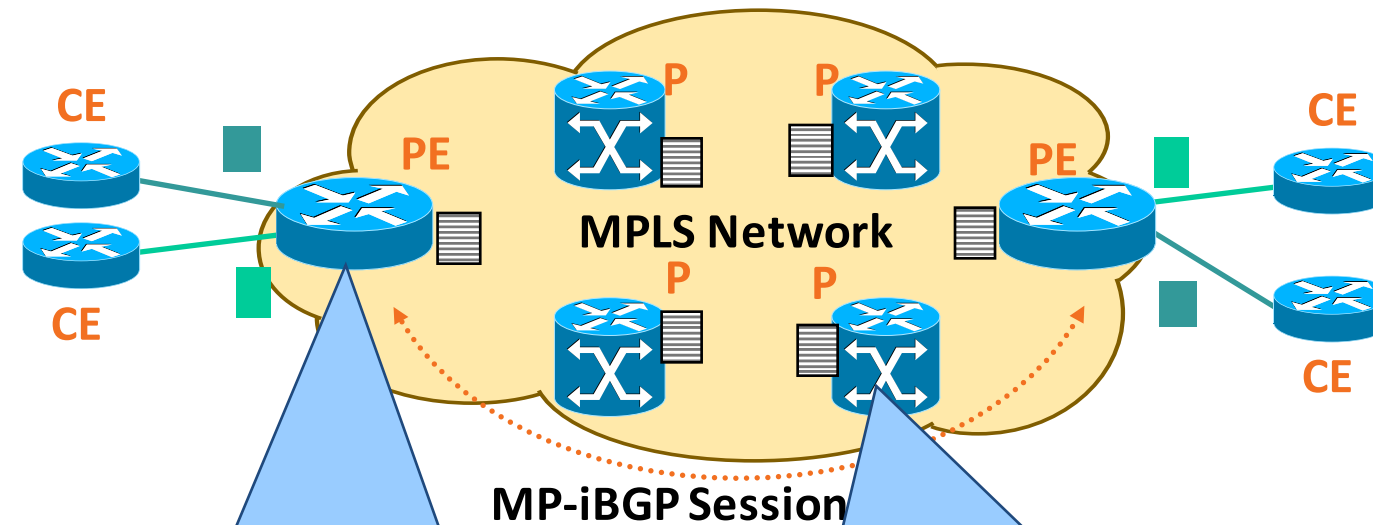
- IP/VPN Overview
  - Technology Overview (How It Works)
  - Configuration Overview
- IP/VPN Services
- Best Practices
- Conclusion

# IP/VPN Technology Overview

- More than one routing and forwarding tables
- Control plane—VPN route propagation
- Data or forwarding plane—VPN packet forwarding

# IP/VPN Technology

## MPLS IP/VPN Topology / Connection Model



### PE Routers

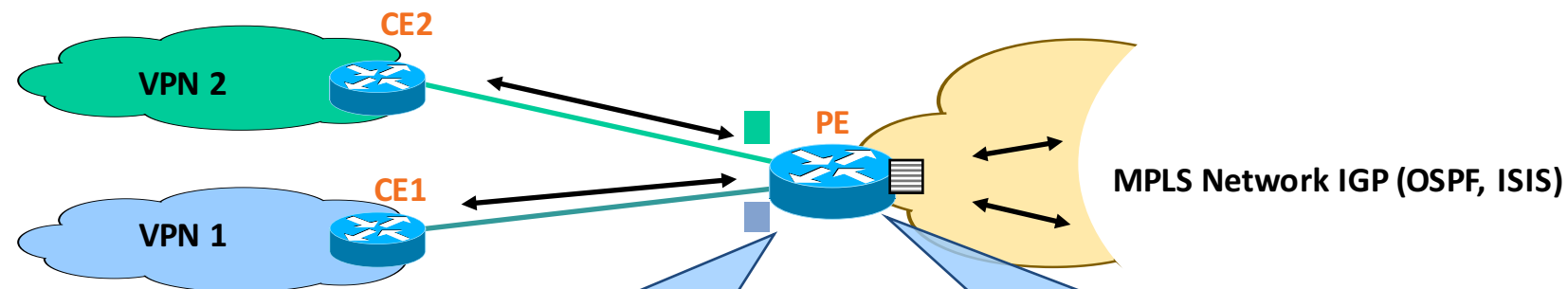
- Sit at the Edge
- Use MPLS with P routers
- Uses IP with CE routers
- Distributes VPN information through MP-BGP to other PE routers

### P Routers

- Sit inside the network
- Forward packets by looking at labels
- P and PE routers share a common IGP

# IP/VPN Technology Overview

## Separate Routing Tables at PE



### Customer Specific Routing Table

- Routing (RIB) and forwarding table (CEF) dedicated to VPN customer
  - VPN1 routing table
  - VPN2 routing table
- Referred to as VRF table for <named VPN>

IOS: "show ip route vrf <name>"  
IOS-XR: "sh route vrf <name> ipv4"  
NX-OS: "sh ip route vrf <name>"

### Global Routing Table

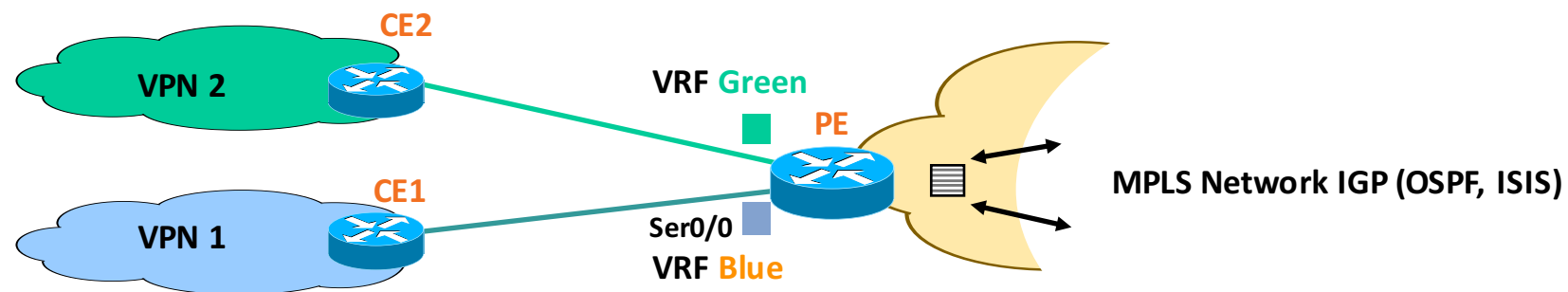
- Created when IP routing is enabled on PE.
- Populated by OSPF, ISIS, etc. running inside the MPLS network

IOS: "show ip route"  
IOS-XR: "sh route ipv4 unicast"  
NX-OS: "sh ip route"



# IP/VPN Technology Overview

## Virtual Routing and Forwarding Instance

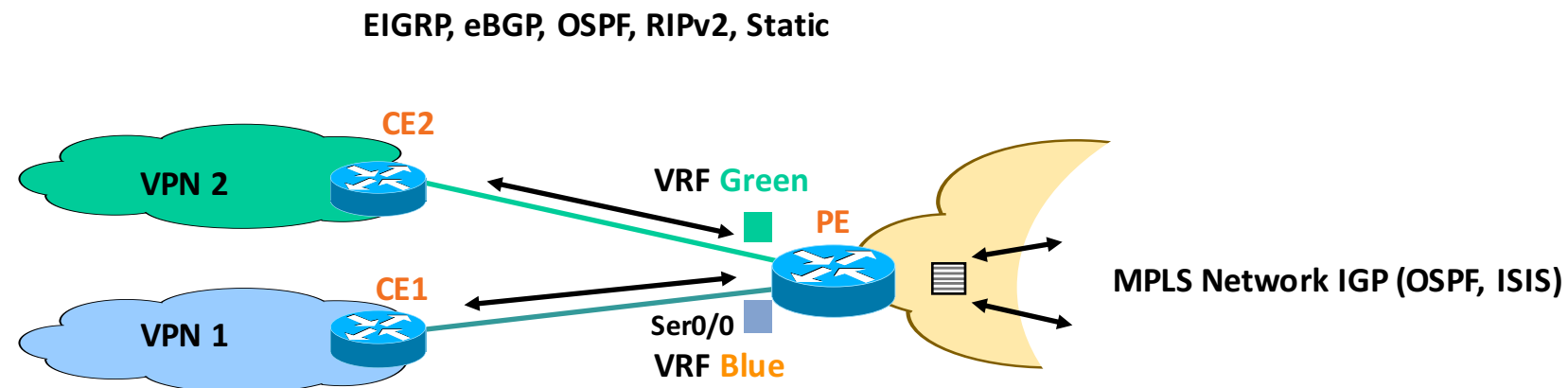


- What's a Virtual Routing and Forwarding (VRF) ?
  - Representation of VPN customer inside the MPLS network
  - Each VPN is associated with at least one VRF
- **VRF configured on each PE** and associated with PE-CE interface(s)
  - Privatize an interface, i.e., coloring of the interface
- **No changes needed at CE**

```
IOS_PE(conf)#ip vrf blue
IOS_PE(conf)#interface Ser0/0
IOS_PE(conf)#ip vrf forwarding blue
```

# IP/VPN Technology Overview

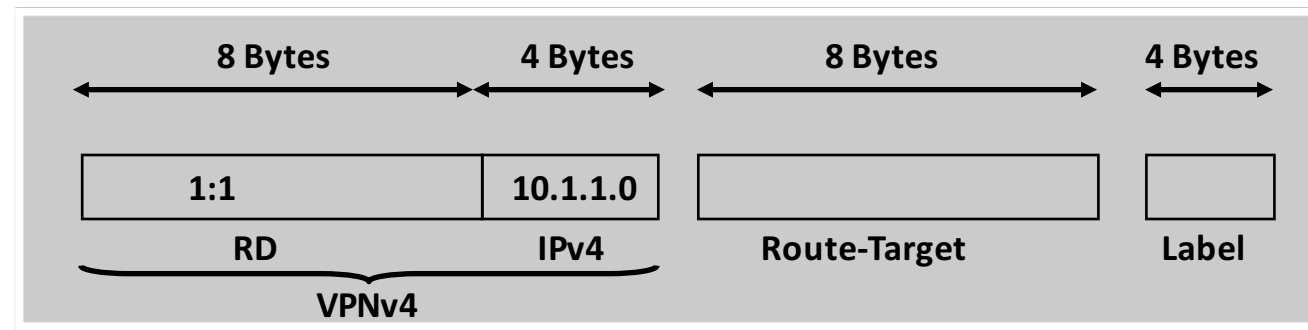
## Virtual Routing and Forwarding Instance



- PE installs the internal routes (IGP) in **global routing table**
- PE installs the VPN customer routes in **VRF routing table(s)**
  - VPN routes are learned from CE routers or remote PE routers
  - VRF-aware routing protocol (static, RIP, BGP, EIGRP, OSPF) on each PE
- **VPN customers can use overlapping IP addresses**
  - BGP plays a key role. Let's understand few BGP specific details.....

# IP/VPN Technology Overview

Control Plane = Multi-Protocol BGP (MP-BGP)



**MP-BGP UPDATE Message**  
Showing VPNv4 Address, RT,  
Label only

MP-BGP Customizes the VPN Customer Routing Information as per the Locally Configured VRF Information at the PE using:

- Route Distinguisher (RD)
- Route Target (RT)
- Label

# IP/VPN Technology Overview: Control Plane

## MP-BGP UPDATE Message Capture

- Visualize how the BGP UPDATE message advertising VPNv4 routes looks like.
- Notice the Path Attributes.

Route Target = 3:3

VPNv4 Prefix 1:1:200.1.62.4/30;  
Label = 23

blackbox desktop (rajiva-u5:1)

File Edit View Capture Analyze Help

No. ,	Time	Source	Destination	Protocol	Info
1	0.000000	10.13.1.5	224.0.0.2	LDP	Hello Message
2	0.350273	10.13.1.5	224.0.0.5	OSPF	Hello Packet
3	102.894345	10.13.1.6	224.0.0.2	LDP	Hello Message
4	103.314144	10.13.1.5	224.0.0.2	LDP	Hello Message
5	103.754579	10.13.1.61	10.13.1.62	BGP	ROUTE-REFRESH Message
6	103.824525	10.13.1.62	10.13.1.61	BGP	UPDATE Message
7	104.054517	10.13.1.61	10.13.1.62	TCP	11002 > 179 [ACK] Seq=23 Ack=91 Win=16274 Len=0
8	104.064465	10.13.1.62	10.13.1.61	BGP	UPDATE Message, UPDATE Message, UPDATE Message
9	104.254411	aa:bb:cc:00:01:00	aa:bb:cc:00:01:00	LOOP	Loopback

Frame 6 (145 bytes on wire, 145 bytes captured)

Ethernet II, Src: aa:bb:cc:00:65:00, Dst: aa:bb:cc:00:01:00

Internet Protocol, Src Addr: 10.13.1.62 (10.13.1.62), Dst Addr: 10.13.1.61 (10.13.1.61)

Transmission Control Protocol, Src Port: 179 (179), Dst Port: 11002 (11002), Seq: 0, Ack: 23, Len: 91

Border Gateway Protocol

UPDATE Message

Marker: 16 bytes

Length: 91 bytes

Type: UPDATE Message (2)

Unfeasible routes length: 0 bytes

Total path attribute length: 68 bytes

Path attributes

ORIGIN: INCOMPLETE (4 bytes)

AS\_PATH: empty (3 bytes)

MULTI\_EXIT\_DISC: 0 (7 bytes)

LOCAL\_PREF: 100 (7 bytes)

EXTENDED\_COMMUNITIES: (11 bytes)

Flags: 0xc0 (Optional, Transitive, Complete)

Type code: EXTENDED\_COMMUNITIES (16)

Length: 8 bytes

Carried Extended communities

Optional, Transitive, CompleteRoute Target: 3:3

MP\_REACH\_NLRI (36 bytes)

Flags: 0x80 (Optional, Non-transitive, Complete)

Type code: MP\_REACH\_NLRI (14)

Length: 33 bytes

Address family: IPv4 (1)

Subsequent address family identifier: Labeled VPN Unicast (128)

Next hop network address (12 bytes)

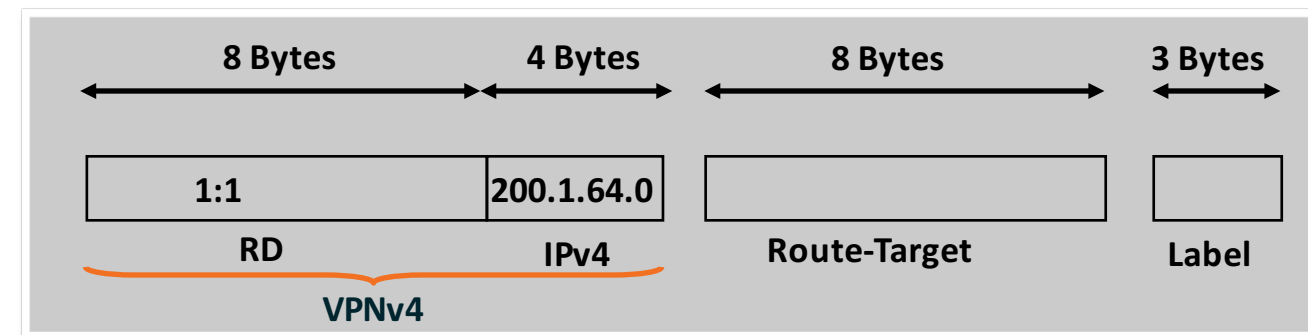
Subnetwork points of attachment: 0

Network layer reachability information (16 bytes)

Label Stack=23 (bottom) RD=1:1, IP=200.1.62.4/30

# IP/VPN Technology Overview: Control Plane

## Route-Distinguisher (rd)



**MP-BGP UPDATE Message  
Showing VPNv4 Address, RT,  
Label only**

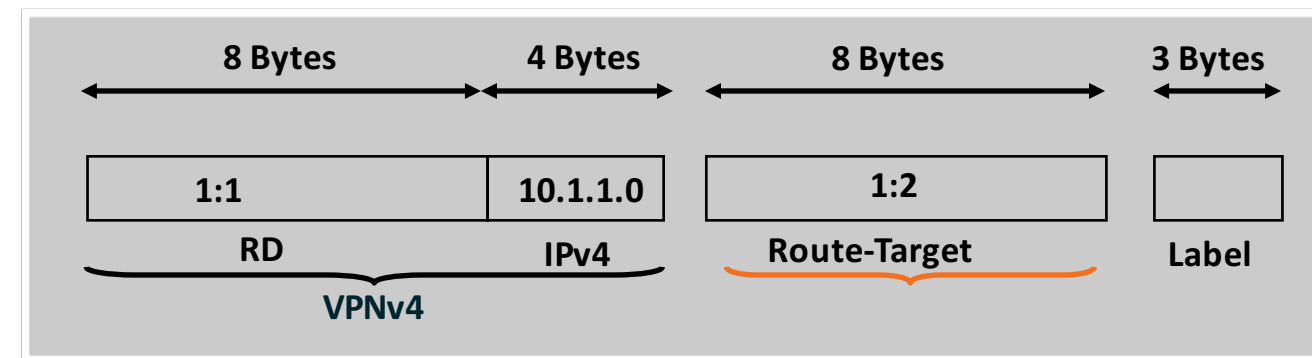
- VPN customer IPv4 prefix is **converted into a VPNv4 prefix** by appending the RD (1:1, say) to the IPv4 address (200.1.64.0, say) => 1:1:200.1.64.0
  - Makes the customer's IPv4 address unique inside the SP MPLS network.
- Route Distinguisher (rd) is configured in the VRF at PE
  - RD is not a BGP attribute, just a field.

```
IOS_PE#  
!  
ip vrf green  
rd 1:1  
!
```

\* After 12.4(3)T, 12.4(3) 12.2(32)S, 12.0(32)S etc., RD Configuration within VRF Has Become **Optional**. Prior to That, It Was Mandatory.

# IP/VPN Technology Overview: Control Plane

## Route-Target (rt)

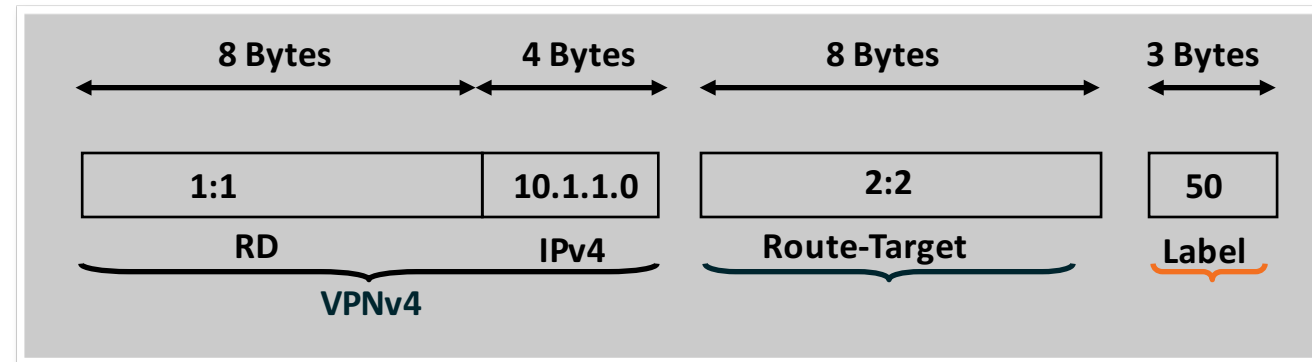


- Route-target (rt) identifies which VRF(s) keep which VPN prefixes
  - rt is an 8-byte extended community attribute.
- Each VRF is configured with a set of route-targets at PE
  - Export and Import route-targets must be the same for any-to-any IP/VPN
- Export route-target values are attached to VPN routes in PE->PE MP-iBGP advertisements

```
IOS_PE#  
!  
ip vrf green  
route-target import 3:3  
route-target export 3:3  
route-target export 10:3  
!
```

# IP/VPN Technology Overview: Control Plane

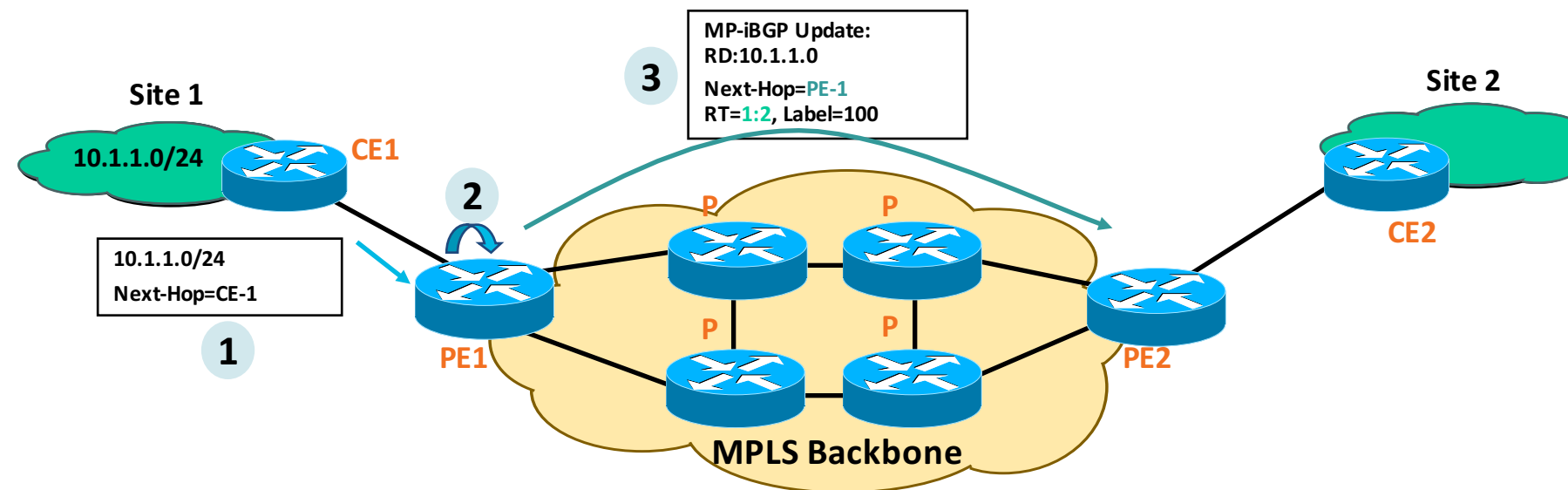
## Label



- PE assigns a label for the VPNv4 prefix;
  - **Next-hop-self towards MP-iBGP neighbors by default** i.e. PE sets the NEXT-HOP attribute to its own address (loopback)
  - Label is not an attribute.
- PE addresses used as BGP next-hop must be uniquely known in IGP
  - **Do not summarize the PE loopback addresses in the core**

# IP/VPN Technology Overview: Control Plane

## Putting it all together

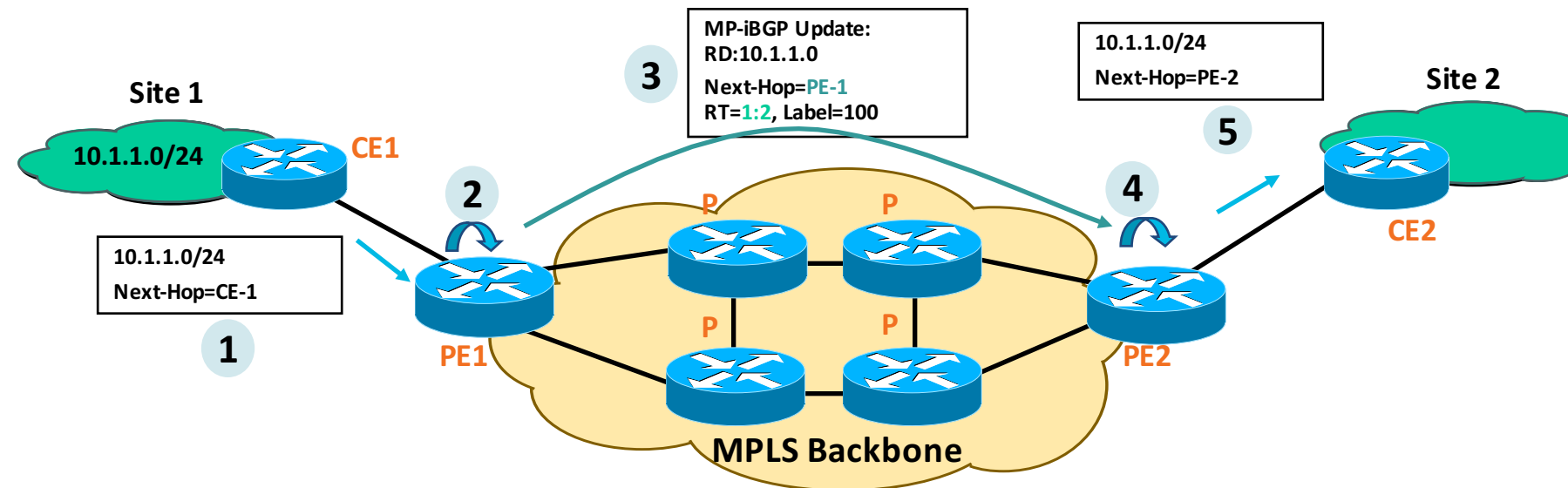


- PE1 receives an IPv4 update (eBGP/OSPF/ISIS/RIP/EIGRP)
- PE1 translates it into VPNv4 address and constructs the MP-iBGP UPDATE message
  - Associates the RT values (export RT =1:2, say) per VRF configuration
  - Rewrites next-hop attribute to itself
  - Assigns a label (100, say); Installs it in the MPLS forwarding table.
- PE1 sends MP-iBGP update to other PE routers



# IP/VPN Technology Overview: Control Plane

Putting it all together

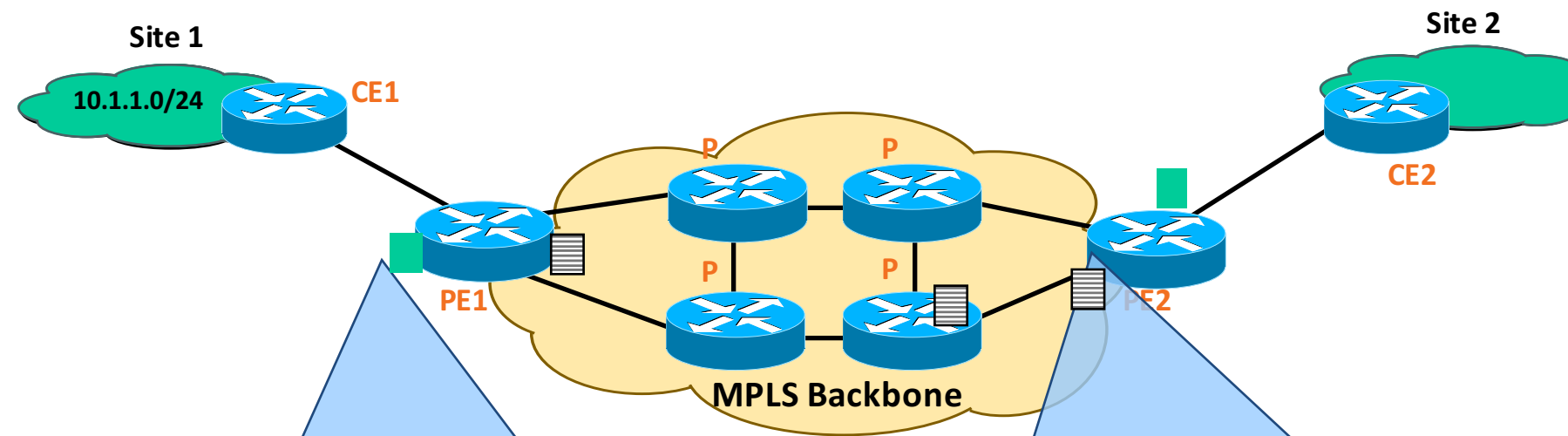


- PE2 receives and checks whether the RT=1:2 is locally configured as 'import RT' within any VRF, if yes, then
  - PE2 translates VPNv4 prefix back to IPv4 prefix
  - Updates the VRF CEF Table for 10.1.1.0/24 with label=100
- PE2 advertises this IPv4 prefix to CE2 (using whatever routing protocol)

**Control Plane is now ready**

# IP/VPN Technology Overview

## Forwarding Plane



### Customer Specific Forwarding Table

- Stores VPN routes with associated labels
- VPN routes learned via BGP
- Labels learned via BGP

IOS:show ip cef vrf <name>

NX-OS: show forwarding vrf <name>

IOS-XR: show cef vrf <name> ipv4

### Global Forwarding Table

- Stores next-hop i.e. PE routes with associated labels
- Next-hop i.e. PE routes learned through IGP
- Label learned through LDP or RSVP

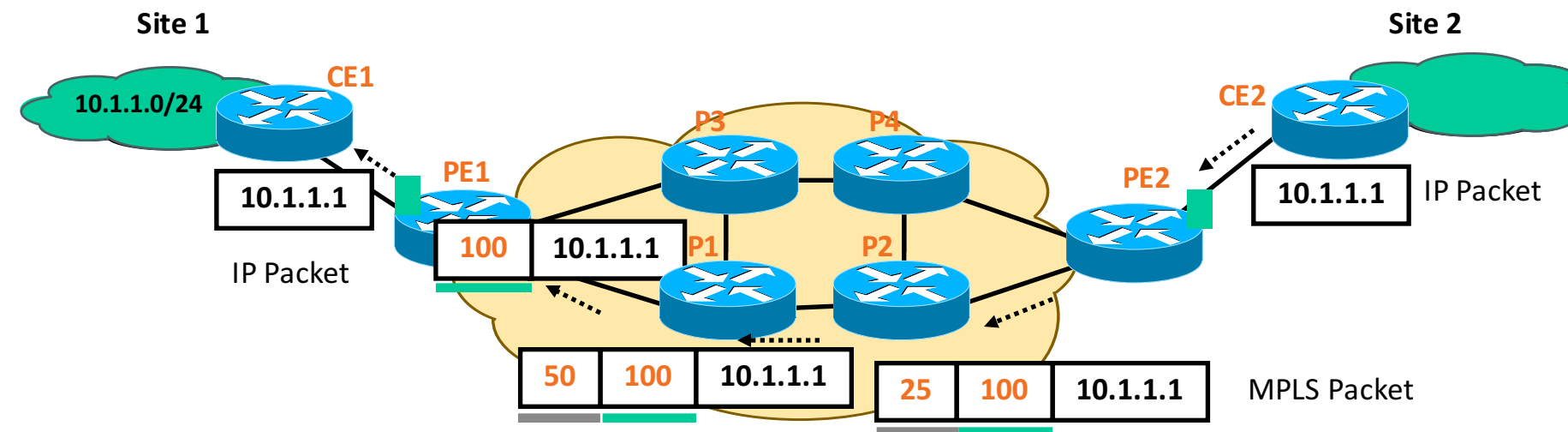
IOS:show ip cef

NX-OS: show forwarding ipv4

IOS-XR: show cef ipv4

# IP/VPN Technology Overview: Forwarding Plane

## Packet Forwarding



- PE2 imposes two labels (MPLS headers) for each IP packet going to site2
  - Outer label is learned via LDP; Corresponds to PE1 address (e.g. IGP route)
  - Inner label is learned via BGP; corresponds to the VPN address (BGP route)
- P1 does the Penultimate Hop Popping (PHP)
- PE1 retrieves IP packet (from received MPLS packet) and forwards it to CE1.

# IP/VPN Technology: Forwarding Plane

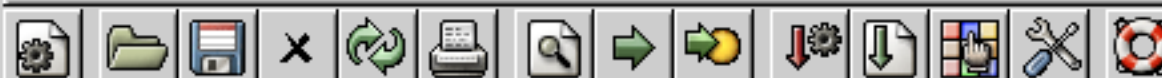
## MPLS IP/VPN Packet Capture

- This capture might be helpful if you never captured an MPLS packet before.

Ethernet Header  
Outer Label

Inner Label

IP Packet

Reference					
<capture> - Ethereal					
File Edit View Capture Analyze Help					
					
No. .	Time	Source	Destination	Protocol	Info
1	0.000000	10.13.1.6	224.0.0.5	OSPF	Hello Packet
2	2.539974	10.13.1.5	224.0.0.5	OSPF	Hello Packet
3	2.870013	10.13.1.5	224.0.0.2	LDP	Hello Message
4	75.051378	10.13.1.6	224.0.0.2	LDP	Hello Message
5	75.190654	aa:bb:cc:00:65:00	aa:bb:cc:00:65:00	LOOP	Loopback
6	75.650449	10.13.1.5	224.0.0.2	LDP	Hello Message
7	77.765333	217.2.61.5	200.1.62.5	ICMP	Echo (ping) request
8	77.798336	217.2.61.5	200.1.62.5	ICMP	Echo (ping) request

⊞	Frame 7 (122 bytes on wire, 122 bytes captured)
⊞	Ethernet II, Src: aa:bb:cc:00:01:00, Dst: aa:bb:cc:00:65:00
⊞	MultiProtocol Label Switching Header
	MPLS Label: Unknown (2003)
	MPLS Experimental Bits: 0
	MPLS Bottom Of Label Stack: 0
	MPLS TTL: 255
⊞	MultiProtocol Label Switching Header
	MPLS Label: Unknown (115)
	MPLS Experimental Bits: 0
	MPLS Bottom Of Label Stack: 1
	MPLS TTL: 255
⊞	Internet Protocol, Src Addr: 217.2.61.5 (217.2.61.5), Dst Addr: 200.1.62.5 (200.1.62.5)
⊞	Internet Control Message Protocol

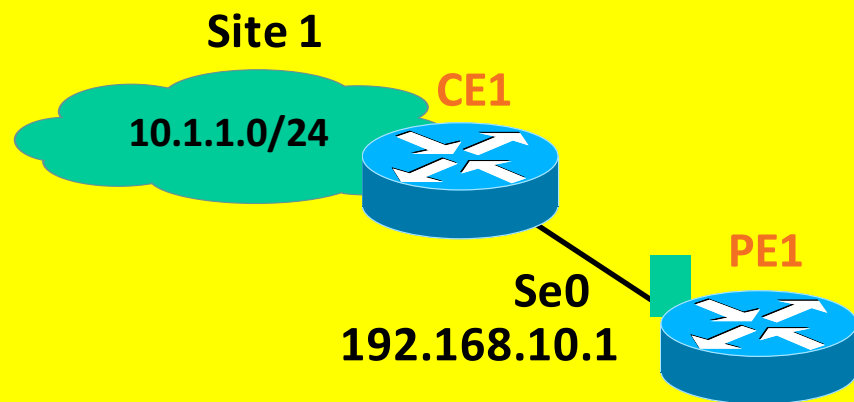
# Agenda

- IP/VPN Overview
  - Technology Overview
  - – Configuration Overview (IOS, IOS-XR and NX-OS)
- IP/VPN Services
- Best Practices
- Conclusion

# MPLS based IP/VPN Sample Configuration (IOS)



## VRF Definition



PE1

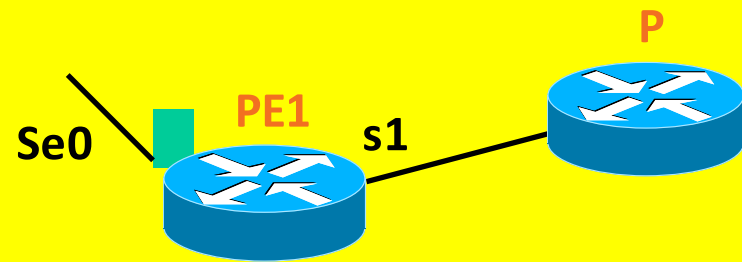
```
ip vrf VPN-A
rd 1:1
route-target export 100:1
route-target import 100:1
```

```
interface Serial0
ip address 192.168.10.1/24
ip vrf forwarding VPN-A
```

```
vrf definition VPN-A
rd 1:1
address-family ipv4
route-target export 100:1
route-target import 100:1
```

```
interface Serial0
ip address 192.168.10.1/24
vrf forwarding VPN-A
```

## PE-P Configuration



PE1

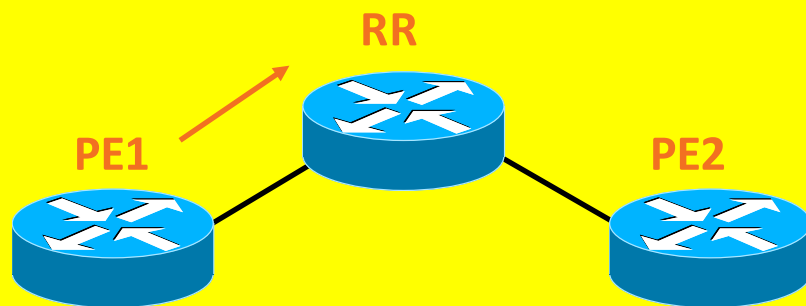
```
Interface Serial1
ip address 130.130.1.1 255.255.255.252
mpls ip
```

```
router ospf 1
network 130.130.1.0 0.0.0.3 area 0
```

# MPLS based IP/VPN Sample Configuration (IOS)



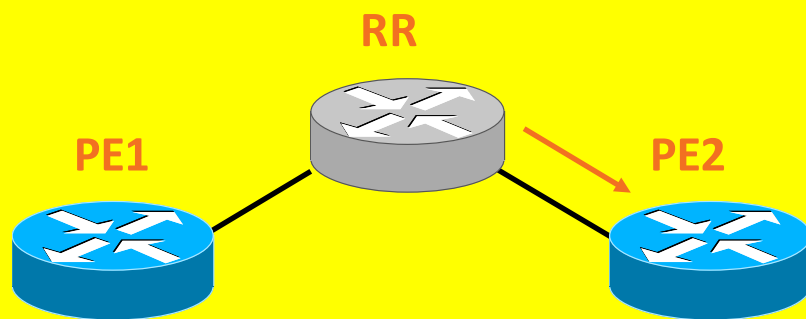
## PE: MP-IBGP Config



PE1

```
router bgp 1
  neighbor 1.2.3.4 remote-as 1
  neighbor 1.2.3.4 update-source loopback0
  !
  address-family vpnv4
    neighbor 1.2.3.4 activate
    neighbor 1.2.3.4 send-community both
  !
```

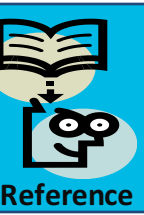
## RR: MP-IBGP Config



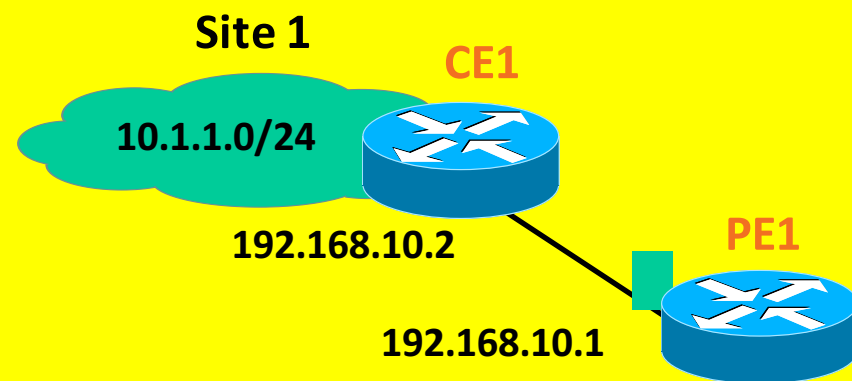
RR

```
router bgp 1
  no bgp default route-target filter
  neighbor 1.2.3.6 remote-as 1
  neighbor 1.2.3.6 update-source loopback0
  !
  address-family vpnv4
    neighbor 1.2.3.6 route-reflector-client
    neighbor 1.2.3.6 activate
  !
```

# MPLS based IP/VPN Sample Configuration (IOS)



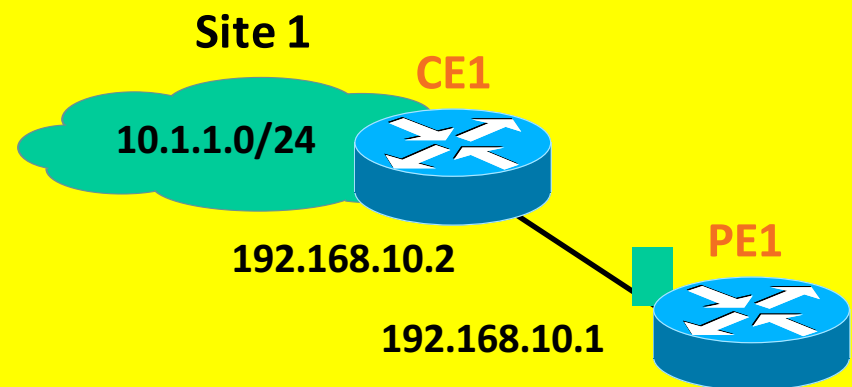
## PE-CE Routing: BGP



PE1

```
router bgp 1
!  
address-family ipv4 vrf VPN-A  
  neighbor 192.168.10.2 remote-as 2  
  neighbor 192.168.10.2 activate  
exit-address-family  
!
```

## PE-CE Routing: OSPF

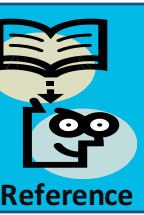


PE1

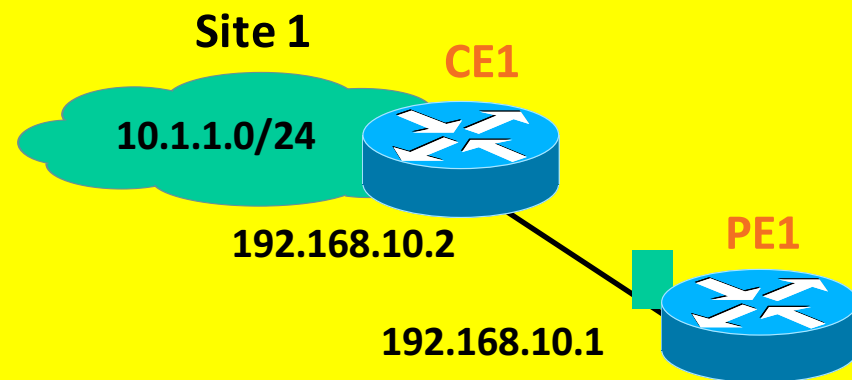
```
router ospf 1
!  
router ospf 2 vrf VPN-A  
  network 192.168.10.0 0.0.0.255 area 0  
  redistribute bgp 1 subnets  
!
```



# MPLS based IP/VPN Sample Configuration (IOS)



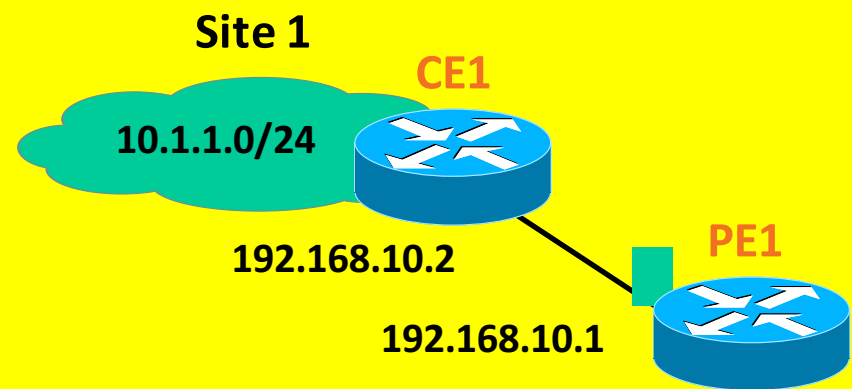
## PE-CE Routing: RIP



PE1

```
router rip
!
address-family ipv4 vrf VPN-A
version 2
no auto-summary
network 192.168.10.0
redistribute bgp 1 metric transparent
!
```

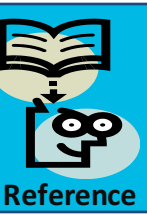
## PE-CE Routing: EIGRP



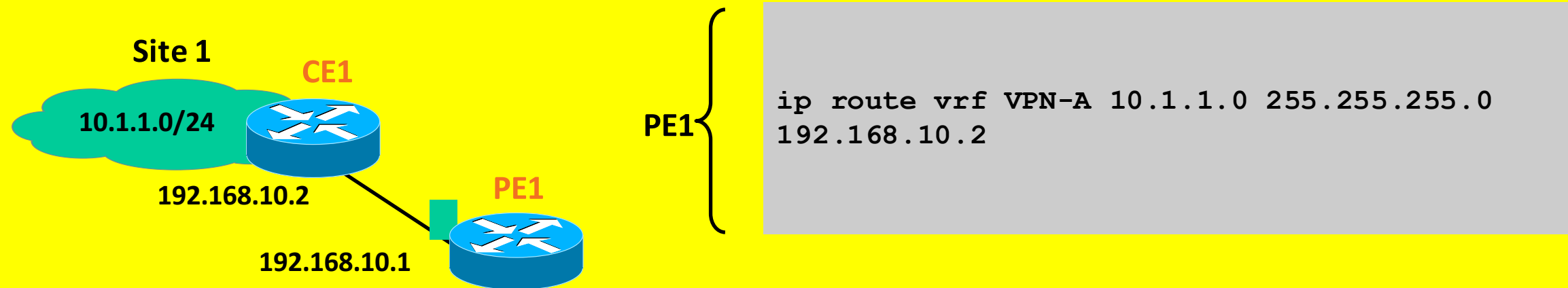
PE1

```
router eigrp 1
!
address-family ipv4 vrf VPN-A
no auto-summary
network 192.168.10.0 0.0.0.255
autonomous-system 10
redistribute bgp 1 metric 100000 100
255 1 1500
!
```

# MPLS based IP/VPN Sample Configuration (IOS)



## PE-CE Routing: Static

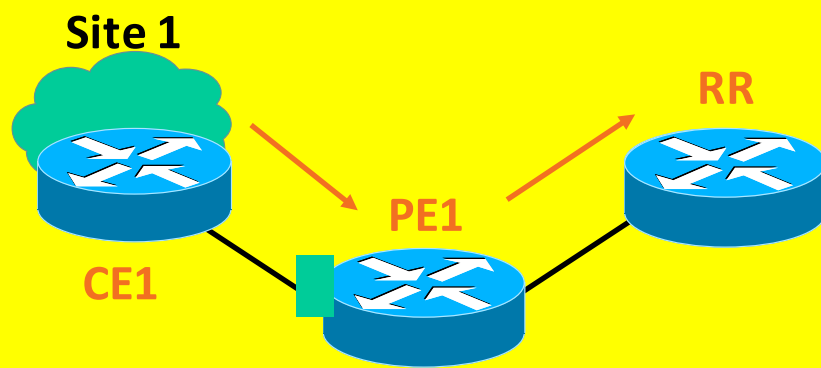


# MPLS based IP/VPN Sample Configuration (IOS)



If PE-CE Protocol Is **Non-BGP**, then Redistribution of Local VPN Routes **into** MP-IBGP Is Required (Shown Below)

## PE-RR (VPN Routes to VPNv4)



PE1

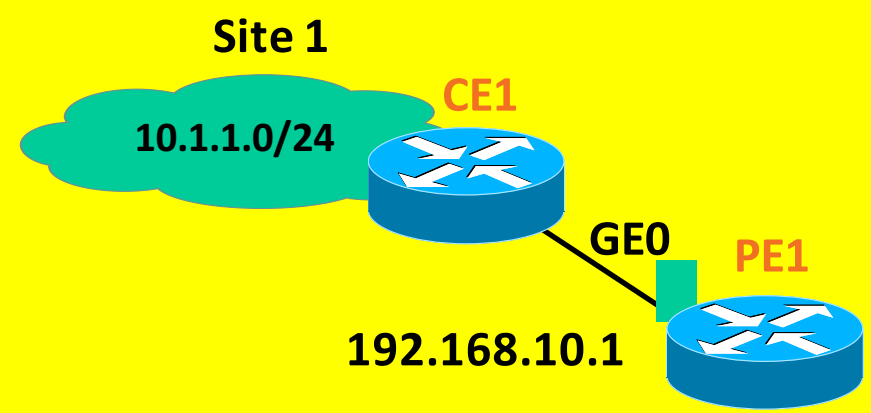
```
router bgp 1
  neighbor 1.2.3.4 remote-as 1
  neighbor 1.2.3.4 update-source loopback 0

  address-family ipv4 vrf VPN-A
    redistribute {rip|connected|static|eigrp|ospf}
```

- Having familiarized with IOS based config, let's peek through IOS-XR and NX-OS config for VPNs

# MPLS based IP/VPN Sample Config (IOS-XR)

## VRF Definition

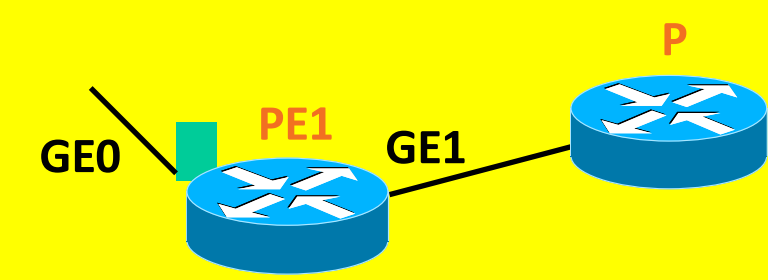


PE1

```
vrf VPN-A
  address-family ipv4 unicast
    import route-target 100:1
    export route-target 100:1
  !
router bgp 1
  vrf VPN-A
    rd 1:1
```

```
Interface GigEthernet0/0/0/0
  ipv4 address 192.168.10.1 255.255.255.0
  vrf VPN-A
```

## PE-P Configuration



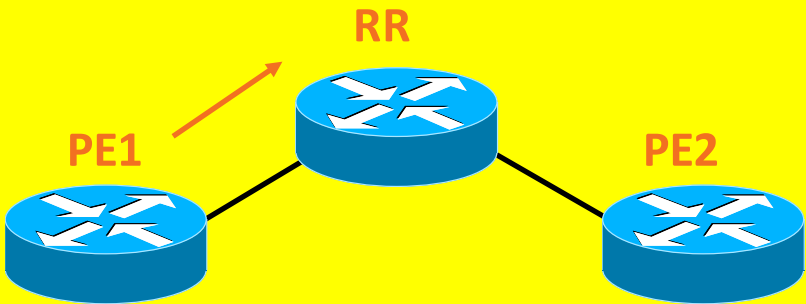
PE1

```
mpls ldp
  route-id 1.2.3.1
  interface GigabitEthernet0/0/0/1
  !
  !
  mpls oam
```

```
router ospf 1
```

# MPLS based IP/VPN Sample Config (IOS-XR)

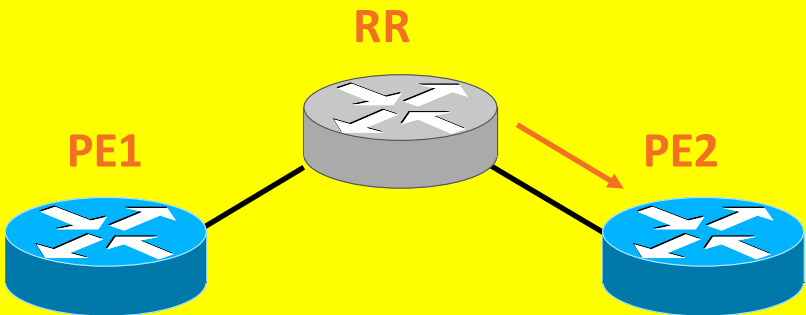
## PE: MP-IBGP Config



PE1

```
router bgp 1
router-id 1.2.3.1
address-family vpnv4 unicast
!
neighbor 1.2.3.4
remote-as 1
update-source loopback0
address-family vpnv4 unicast
!
!
```

## RR: MP-IBGP Config

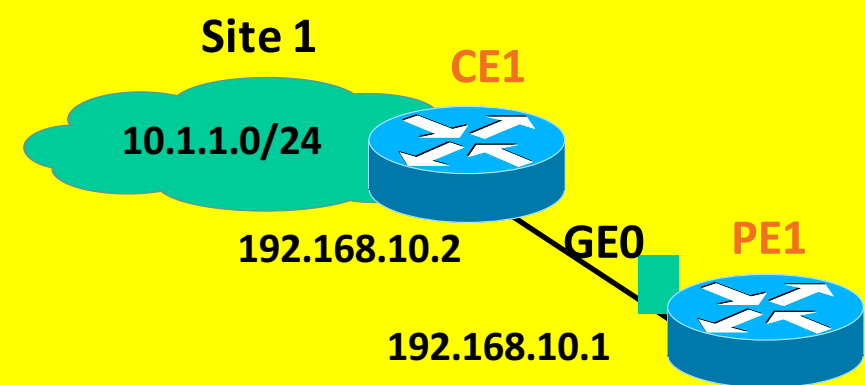


RR

```
router bgp 1
router-id 1.2.3.4
address-family vpnv4 unicast
!
neighbor 1.2.3.1
remote-as 1
update-source loopback0
address-family vpnv4 unicast
route-reflector-client
!
!
```

# MPLS based IP/VPN Sample Config (IOS-XR)

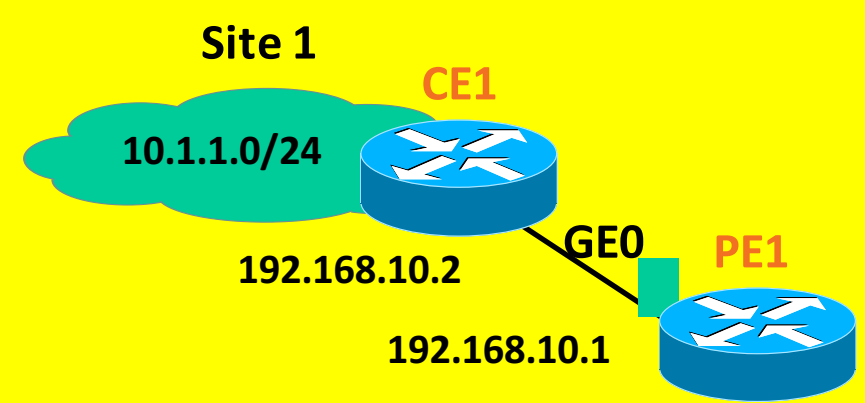
## PE-CE Routing: BGP



PE1

```
router bgp 1
  vrf VPN-A
    address-family ipv4 unicast
      neighbor 192.168.10.2
      remote-as 2
    address-family ipv4 unicast
      route-policy pass-all in|out
  !
  !
  !
  !
```

## PE-CE Routing: OSPF

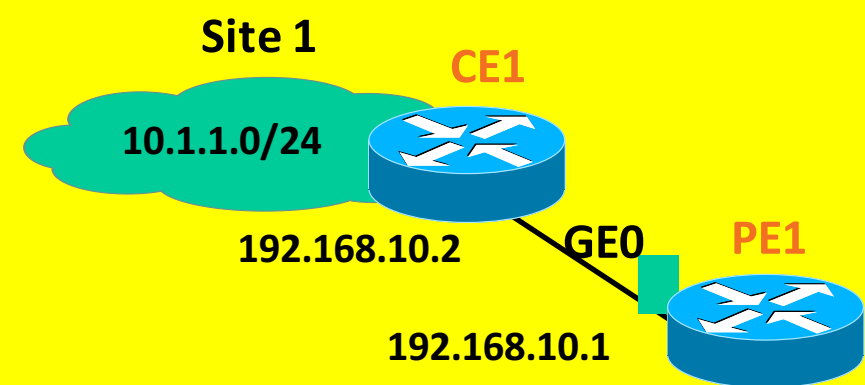


PE1

```
router ospf 2
  vrf VPN-A
    redistribute bgp 1
    area 0
      interface GigabitEthernet0/0/0/1
  !
  !
```

# MPLS based IP/VPN Sample Config (IOS-XR)

## PE-CE Routing: RIP



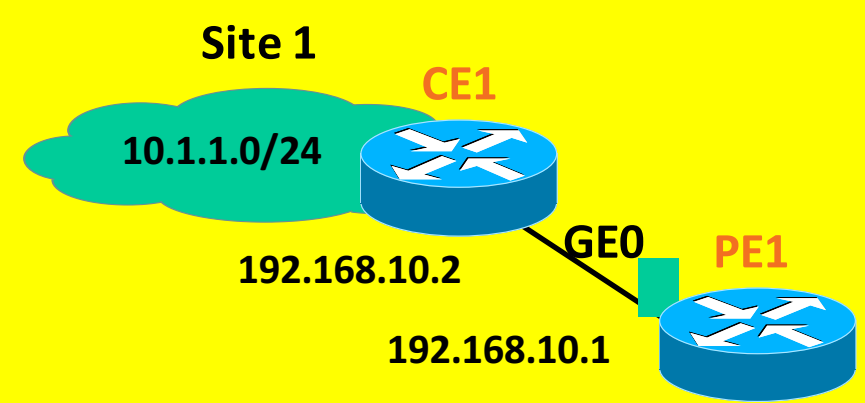
PE1 {

```

router rip
vrf VPN-A
interface GigabitEthernet0/0/0/0
redistribute bgp 1
!

```

## PE-CE Routing: EIGRP



PE1 {

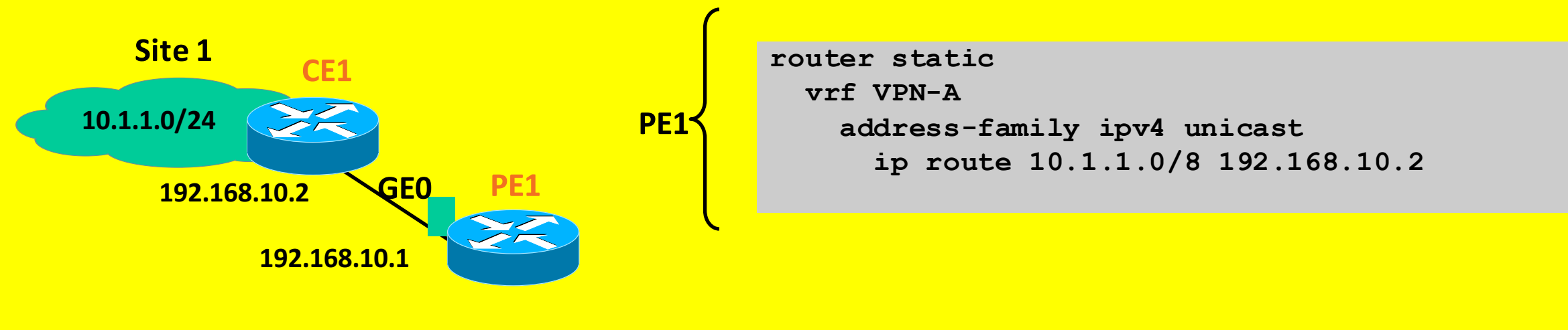
```

router eigrp 1
vrf VPN-A
address-family ipv4
as 10
default-metric 100000 100 255 1 1500
interface GigabitEthernet0/0/0/0
redistribute bgp 1

```

# MPLS based IP/VPN Sample Config (IOS-XR)

## PE-CE Routing: Static

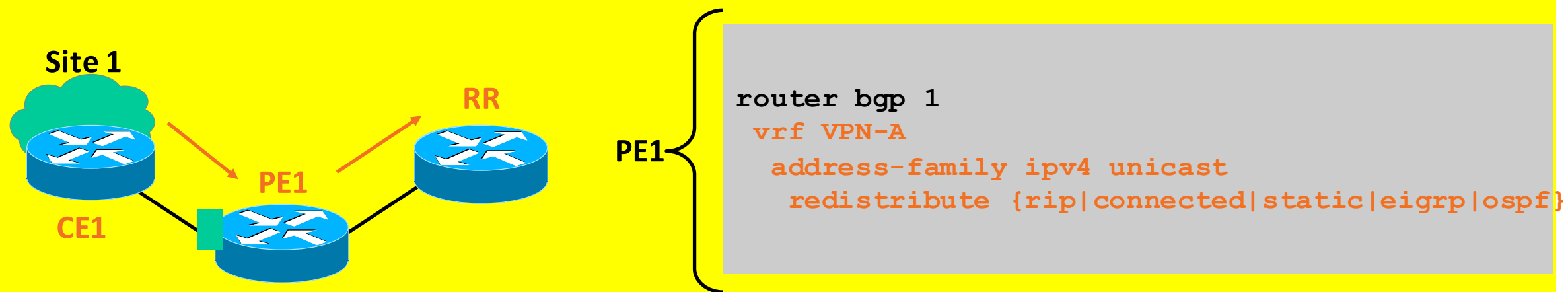




# MPLS based IP/VPN Sample Config (IOS-XR)

If PE-CE Protocol Is **Non-BGP**, then Redistribution of Local VPN Routes **into** MP-IBGP Is Required (Shown Below)

## PE-RR (VPN Routes to VPNv4)

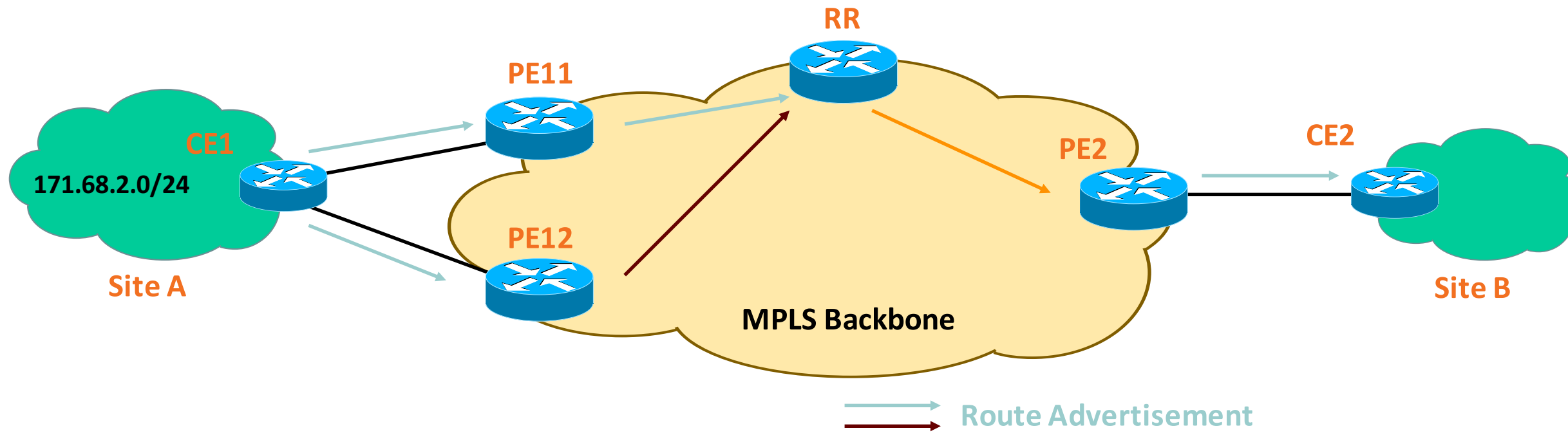


# Agenda

- IP/VPN Overview
- IP/VPN Services
  - 1. **Load-Sharing for Multihomed VPN Sites**
  - 2. Hub and Spoke Service
  - 3. Extranet Service
  - 4. Internet Access Service
  - 5. IP/VPN over IP Transport
  - 6. IPv6 VPN Service
- Best Practices
- Conclusion

# IP/VPN Services:

## 1. Loadsharing of VPN Traffic

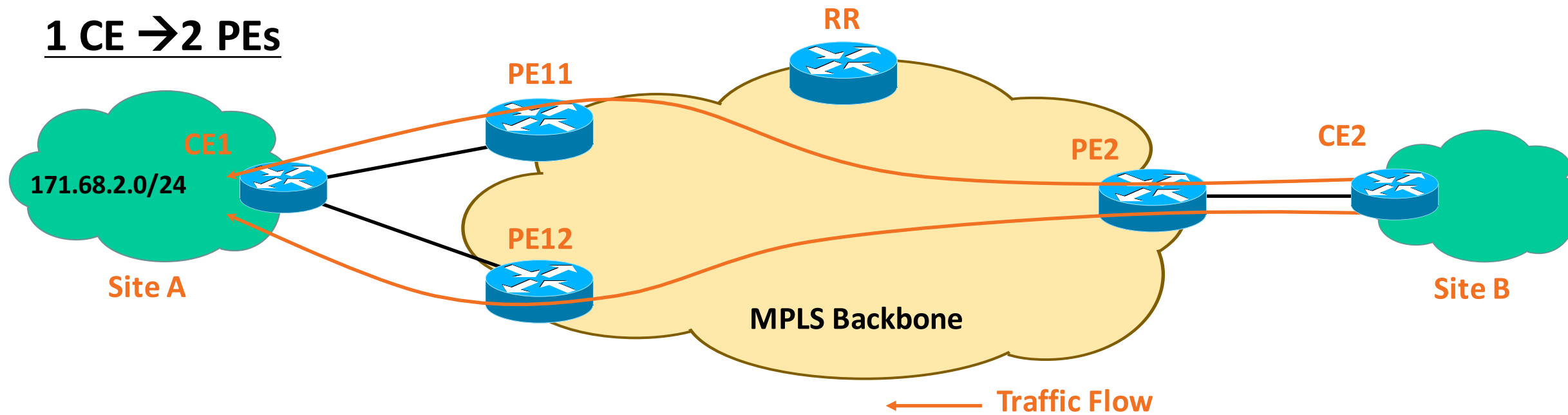


- VPN sites (such as Site A) could be multihomed
- VPN customer may demand the traffic (to the multihomed site) be loadshared

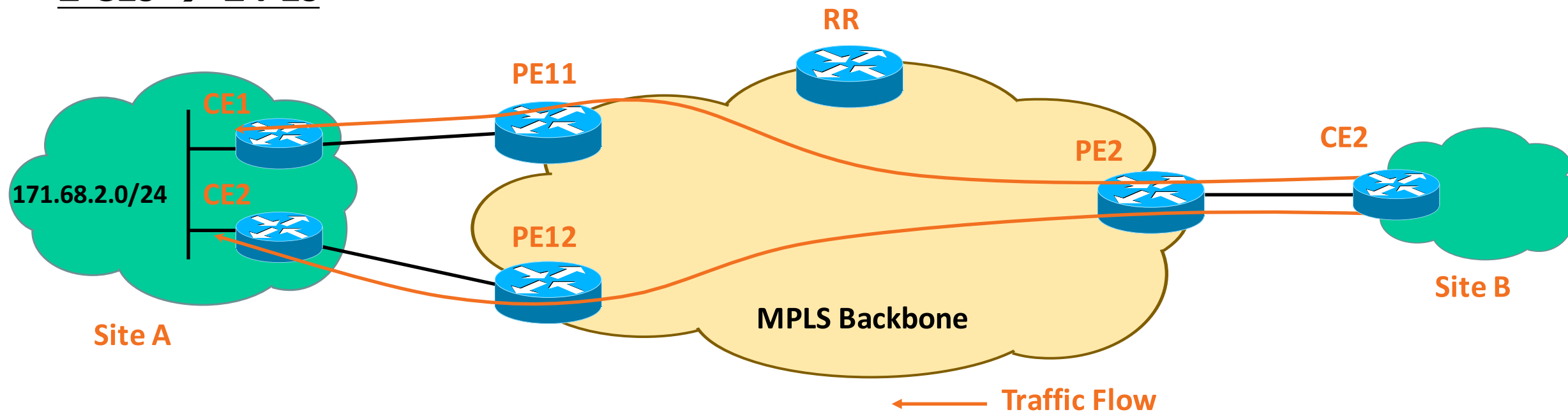
# IP/VPN Services:

## 1. Loadsharing of VPN Traffic: Two Scenarios

### 1 CE → 2 PEs



### 2 CEs → 2 PEs



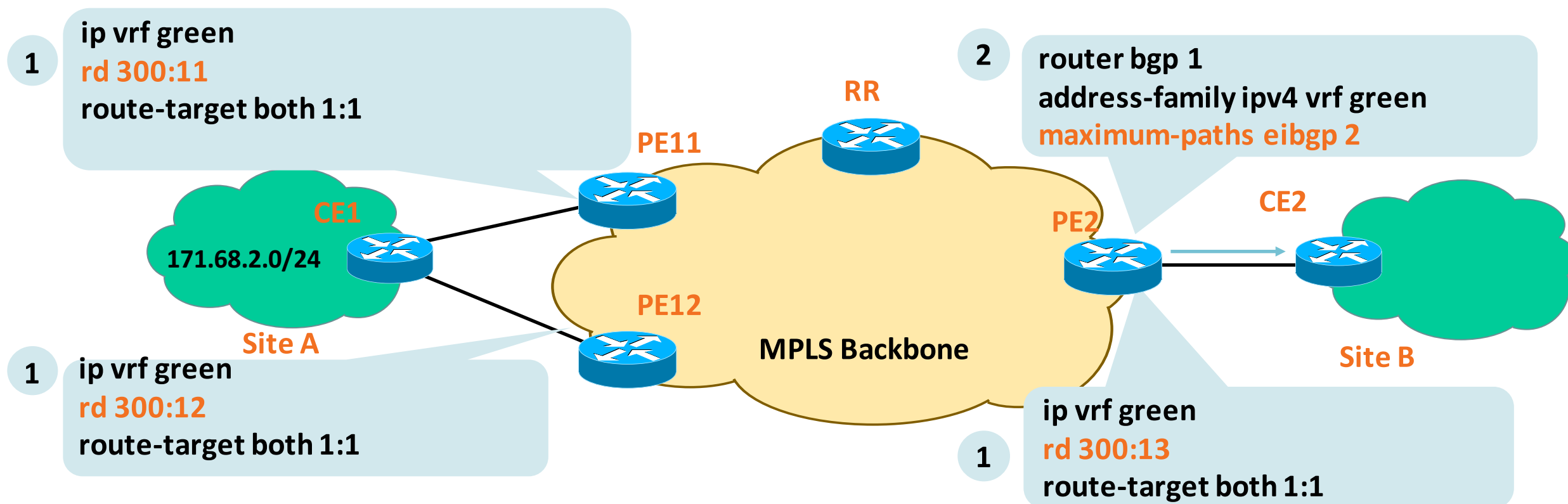
# IP/VPN Services:

## 1. Loadsharing of VPN Traffic: IOS Configuration

Configure unique RD per VRF per PE for multihomed site/interfaces

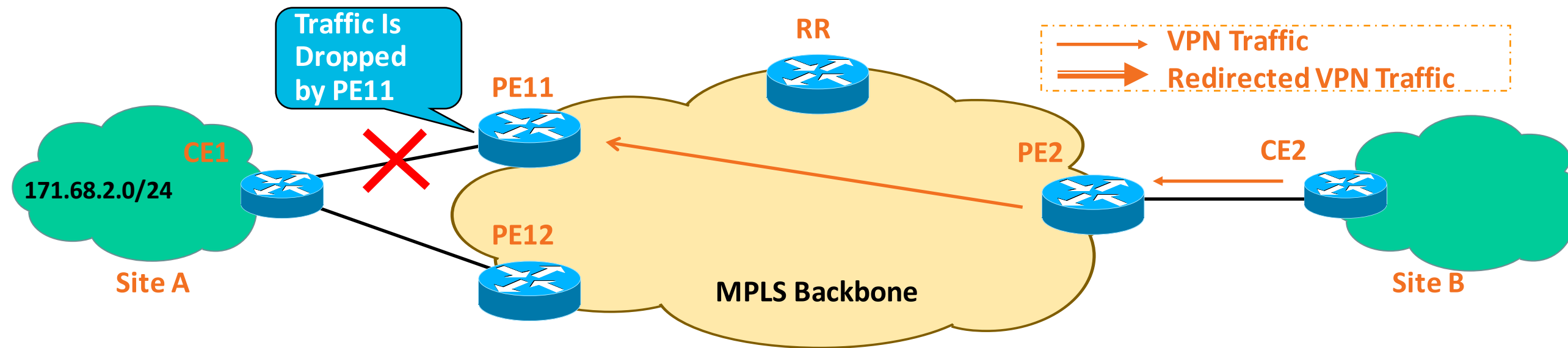
–Assuming RR exists

Enable BGP multipath within the relevant BGP VRF address-family at remote PE routers such as PE2 (why PE2?).



# IP/VPN Services:

## 1. VPN Fast Convergence—PE-CE Link Failure



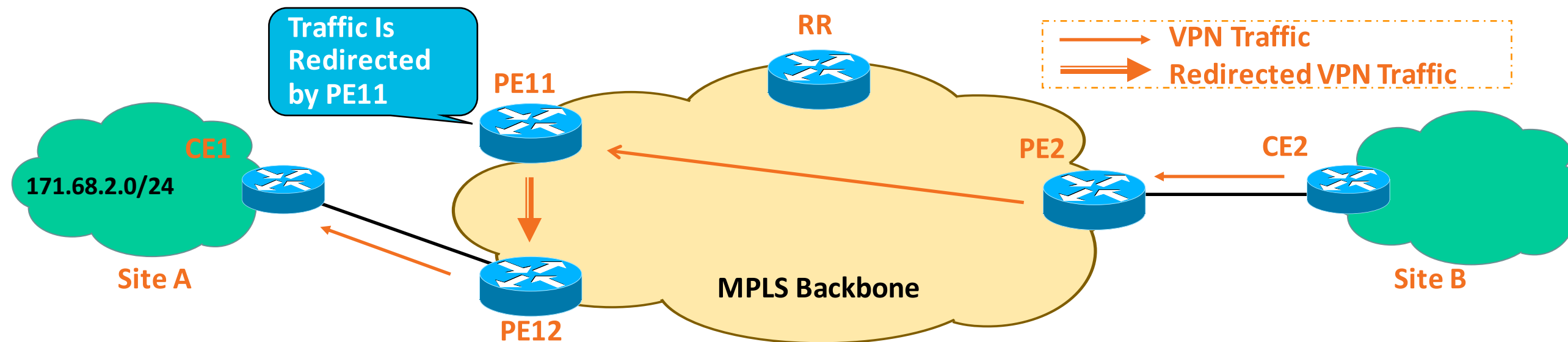
In a classic multi-homing case, PE11, upon detecting the PE-CE link failure, sends BGP message to withdraw the VPN routes towards other PE routers.

–This results in the remote PE routers selecting the alternate bestpath (if any), but until then, they keep sending the MPLS/VPN traffic to PE11, which keeps dropping the traffic.

Use **PIC Edge** feature to minimize the loss due to the PE-CE link failure from sec to msec .

# IP/VPN Services:

## 1. VPN Fast Convergence—PE-CE Link Failure



‘BGP PIC Edge’ feature helps PE11 to minimize the traffic loss from sec to msec, during local PE-CE link failure

- PE11 immediately reprograms the forwarding entry with the alternate BGP best path (which is via PE12)
- PE11 redirects the CE1 bound traffic to PE12 (with the right label)

In parallel, PE11 sends the ‘BGP withdraw message’ to RR/PE2, which will run the bestpath algorithm and removes the path learned via PE11, and then adjust their forwarding entries via PE12

**This feature is independent of whether multipath is enabled on PE2 or not, however, dependent on VPN site multihoming**

# Agenda

- IP/VPN Overview
- IP/VPN Services
  1. Load-Sharing for Multihomed VPN Sites
  2. **Hub and Spoke Service**
  3. Extranet Service
  4. Internet Access Service
  5. IP/VPN over IP Transport
  6. IPv6 VPN Service
- Best Practices
- Conclusion



# IP/VPN Services:

## 2. Hub and Spoke Service

- Many VPN deployments need to be hub and spoke
  - Spoke to spoke communication via Hub site only
- Despite MPLS based IP/VPN's **implicit any-to-any, i.e., full-mesh connectivity**, hub and spoke service can easily be offered
  - Done with **import and export of route-target (RT) values**
  - **Requires unique RD per VRF per PE**
- PE routers can run any routing protocol with VPN customer' hub and spoke sites independently

# IP/VPN Services:

## 2. Hub and Spoke Service

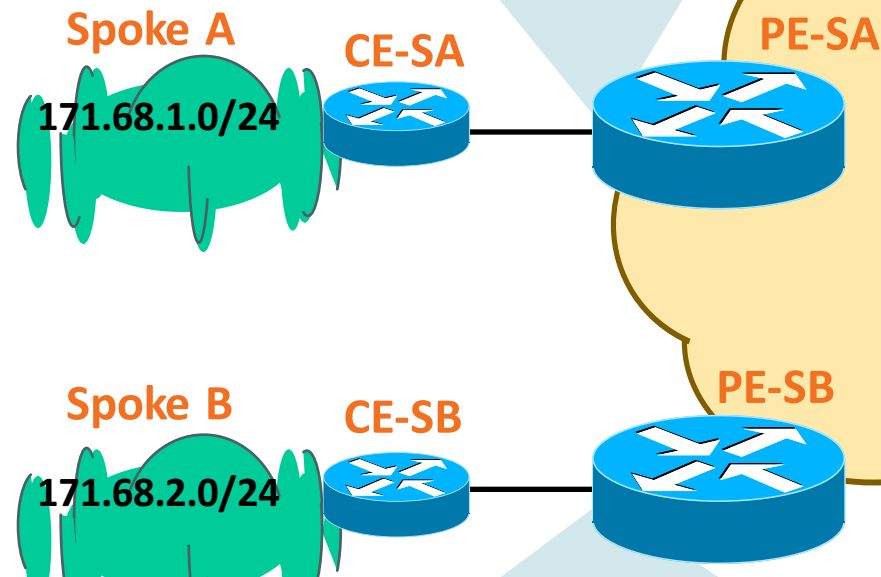
- Two configuration Options :
  1. 1 PE-CE interface to Hub & 1 VRF;
  2. 2 PE-CE interfaces to Hub & 2 VRFs;
- Use **option#1** if Hub site advertises default or summary routes towards the Spoke sites, **otherwise use Option#2**
- HDVRF feature\* allows the option#2 to use just one PE-CE interface

\* HDVRF Feature Is Discussed Later

# IP/VPN Services:

## 2. Hub and Spoke Service: IOS Configuration – Option#1

```
ip vrf green-spoke1
description VRF for SPOKE A
rd 300:111
route-target export 1:1
route-target import 2:2
```



```
ip vrf HUB
description VRF for HUB
rd 300:11
route-target import 1:1
route-target export 2:2
```

PE-Hub

Eth0/0

CE-Hub

MPLS VPN Backbone

- PE-Hub MUST advertise only default or aggregate route(s) to PE-SA/SB
- PE-Hub MUST NOT use bgp aggregation

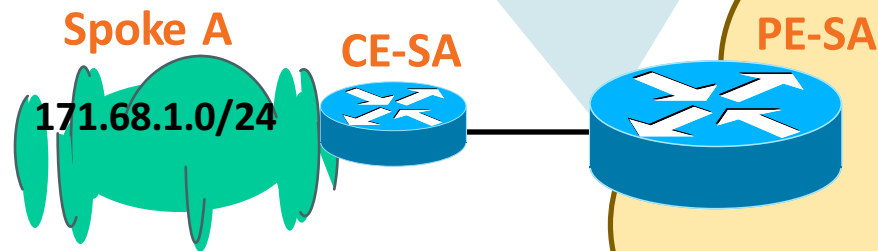
Import and Export RT  
Values Must Be Different

Supported in IOS,  
NXOS and IOS-XR

# IP/VPN Services:

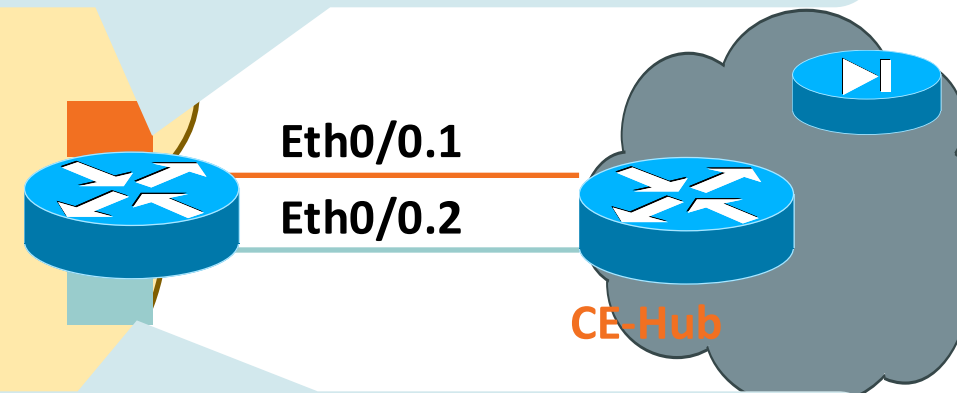
## 2. Hub and Spoke Service: IOS Configuration – Option#2

```
ip vrf green-spoke1
description VRF for SPOKE A
rd 300:111
route-target export 1:1
route-target import 2:2
```



```
ip vrf green-spoke2
description VRF for SPOKE B
rd 300:112
route-target export 1:1
route-target import 2:2
```

```
ip vrf HUB-IN
description VRF for traffic from HUB
rd 300:11
route-target import 1:1
```



```
ip vrf HUB-OUT
description VRF for traffic to HUB
rd 300:12
route-target export 2:2
```

- PE-Hub can advertise Spoke specific route(s) to PE-SA/SB.
- PE-Hub MAY use bgp aggregation.

# IP/VPN Services:

## 2. Hub and Spoke Service: Configuration – Option#2

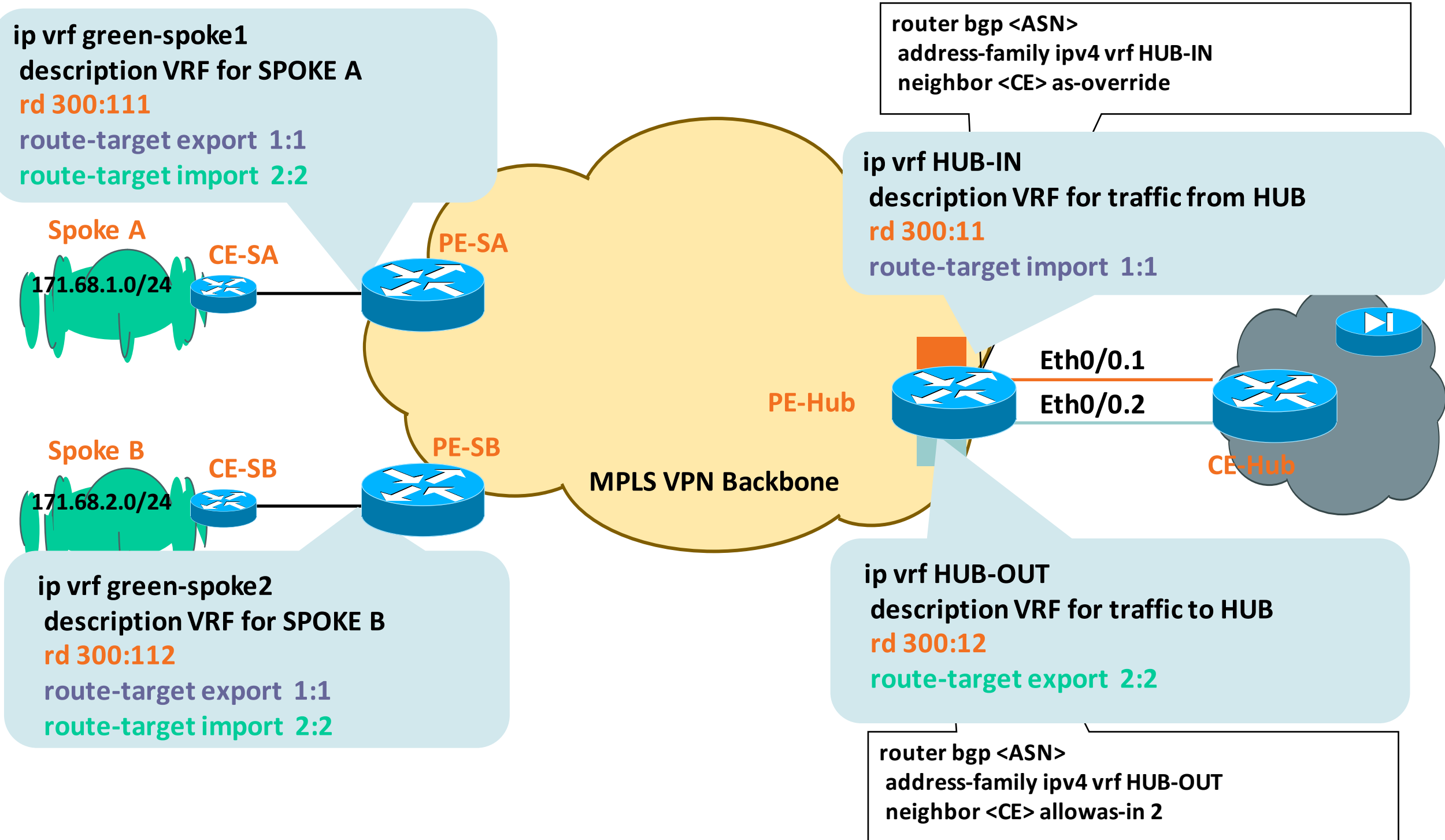
- If BGP is used between every PE and CE, then **allowas-in** and **as-override\*** knobs must be used at the PE\_Hub\*\*
  - Otherwise AS\_PATH looping will occur

\* Only If Hub and Spoke Sites Use the Same BGP ASN

\*\* Configuration for This Is Shown on the Next Slide

# IP/VPN Services:

## 2. Hub and Spoke Service: Configuration – Option#2

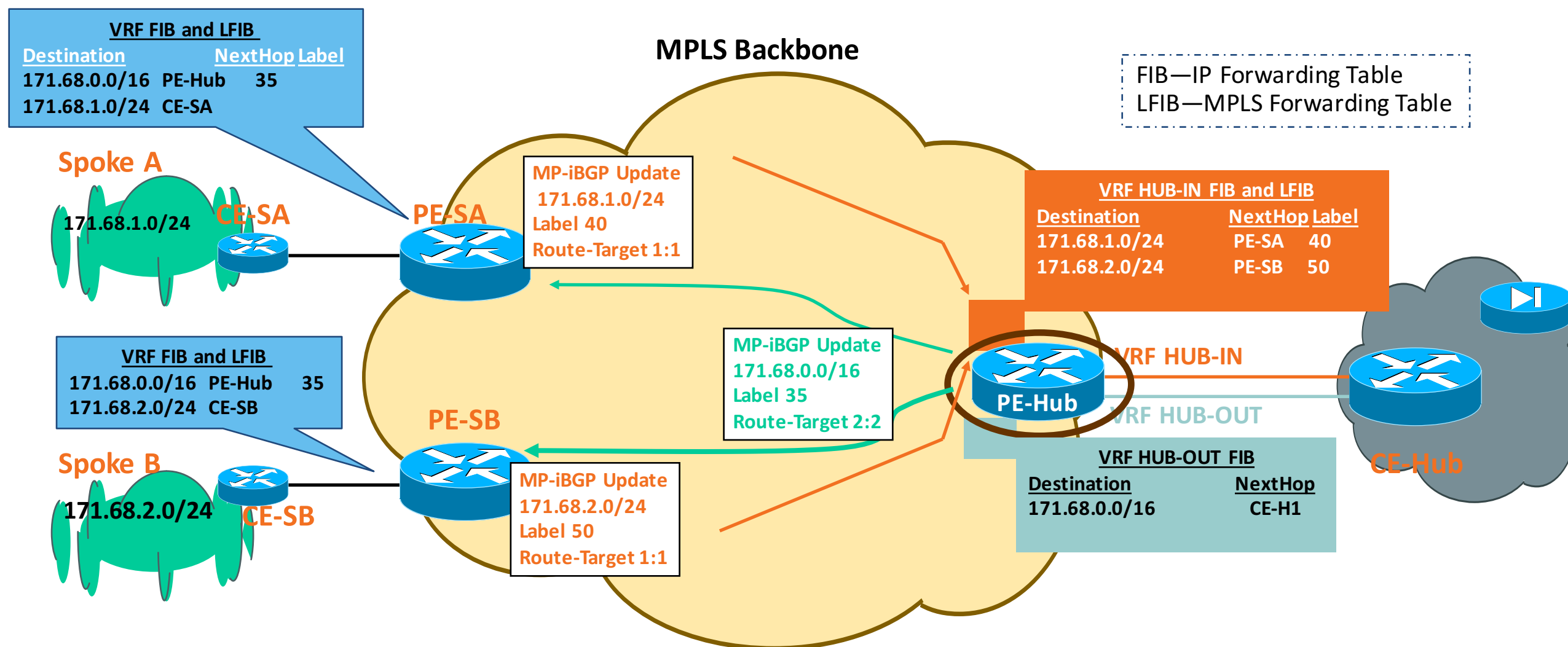


# IP/VPN Services:

## 2. Hub and Spoke Service: Control Plane (Option#2)

### Two VRFs at the PE-Hub:

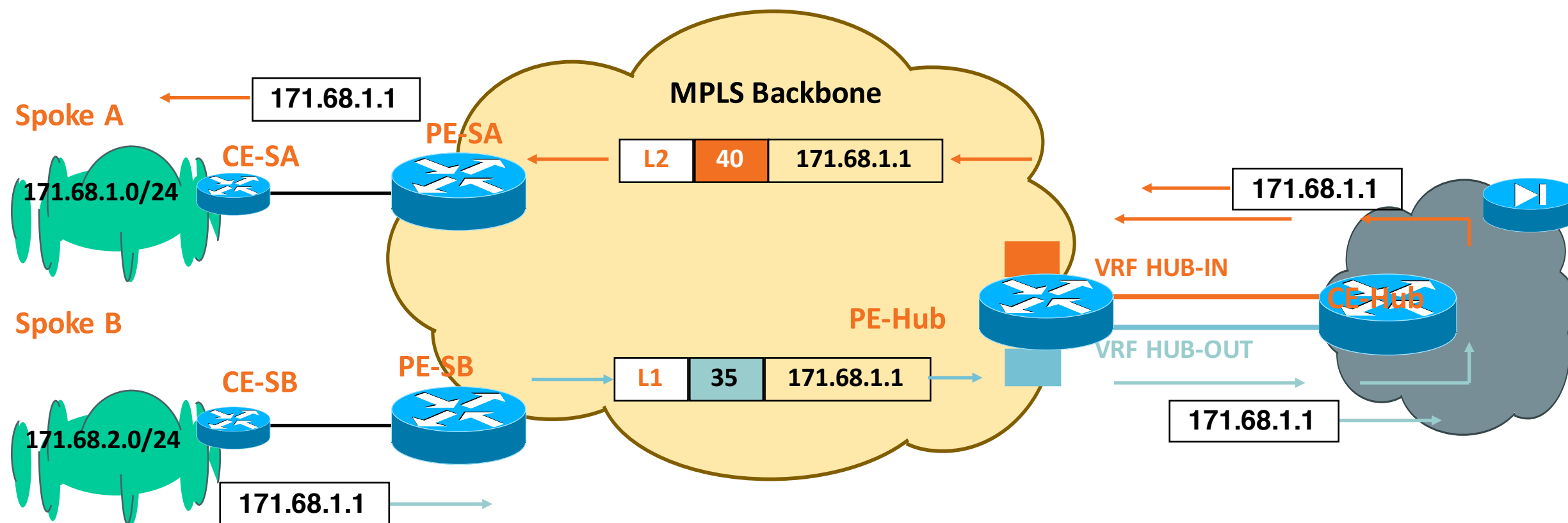
- VRF HUB-IN to learn every spoke routes from remote PEs
- VRF HUB-OUT to advertise spoke routes or summary 171.68.0.0/16 routes to remote PEs



# IP/VPN Services:

## 2. Hub and Spoke Service: Forwarding Plane (Option#2)

This Is How the Spoke-to-Spoke Traffic Flows



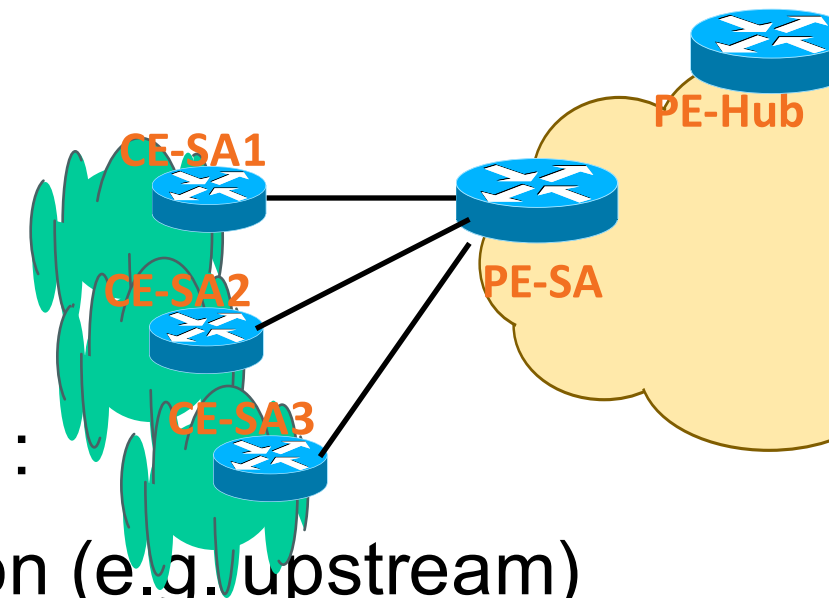
L1 Is the Label to Get to PE-Hub  
L2 Is the Label to Get to PE-SA



# IP/VPN Services:

## 2. What If Many Spoke Sites Connect to the Same PE Router?

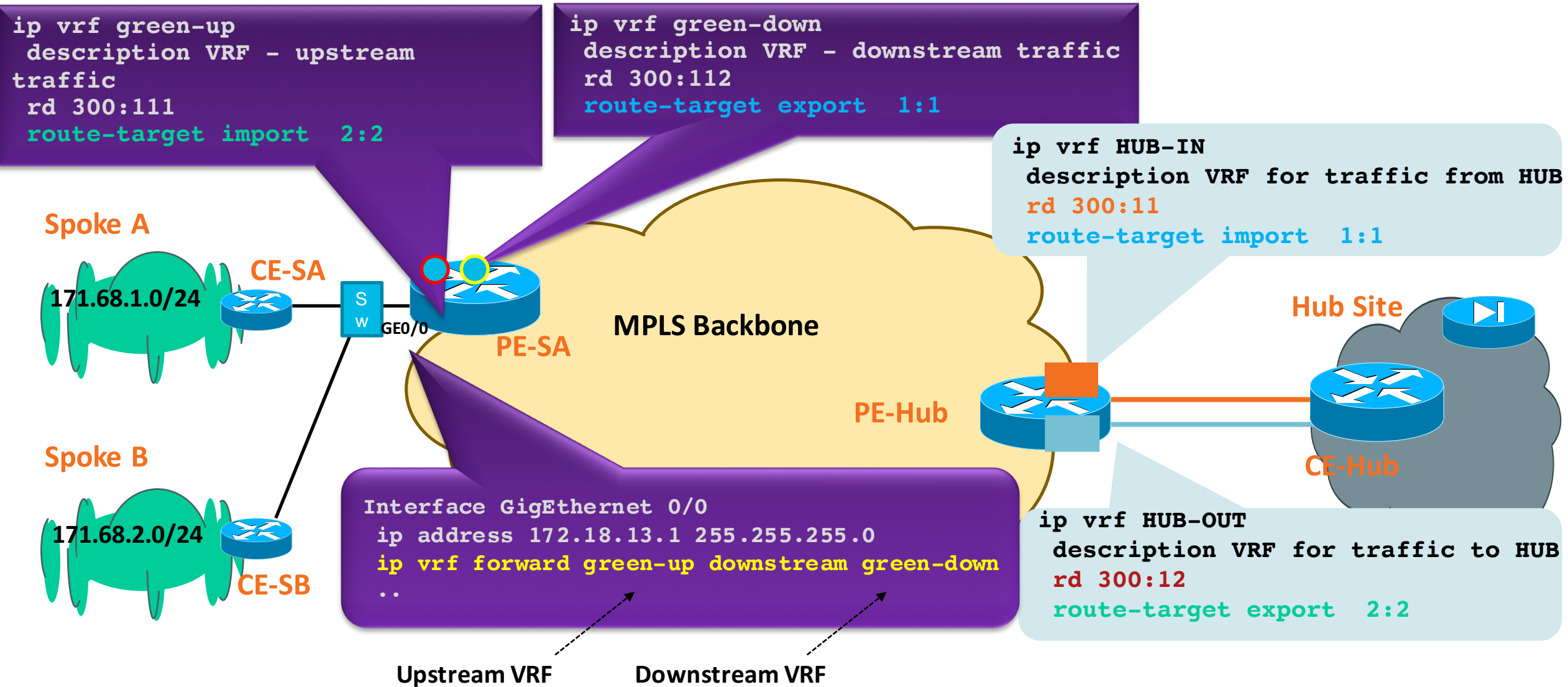
- If more than one spoke router (CE) connects to the same PE router (within the same VRF), then such **spokes can reach other without needing the hub.**
  - Defeats the purpose of hub and spoke ☹️



- **Half-duplex VRF is the answer**
  - Uses two VRFs on the PE (spoke) router :
  - A VRF for spoke->hub communication (e.g. upstream)
  - A VRF for spoke<-hub communication (e.g. downstream)


# IP/VPN Services:

## 2. Hub and Spoke Service: Half-Duplex VRF



1. PE-SA installs the Spoke routes only in downstream VRF i.e. green-down
2. PE-SA installs the Hub routes only in upstream VRF i.e. green-up
3. PE-SA forwards the incoming IP traffic (from Spokes) using upstream VRF i.e. green-up routing table.
4. PE-SA forwards the incoming MPLS traffic (from Hub) using downstream VRF i.e. green-down routing table

# Agenda

- IP/VPN Overview
- IP/VPN Services
  1. Load-Sharing for Multihomed VPN Sites
  2. Hub and Spoke Service
  -  3. **Extranet Service**
  4. Internet Access Service
  5. IP/VPN over IP Transport
  6. IPv6 VPN Service
- Best Practices
- Conclusion

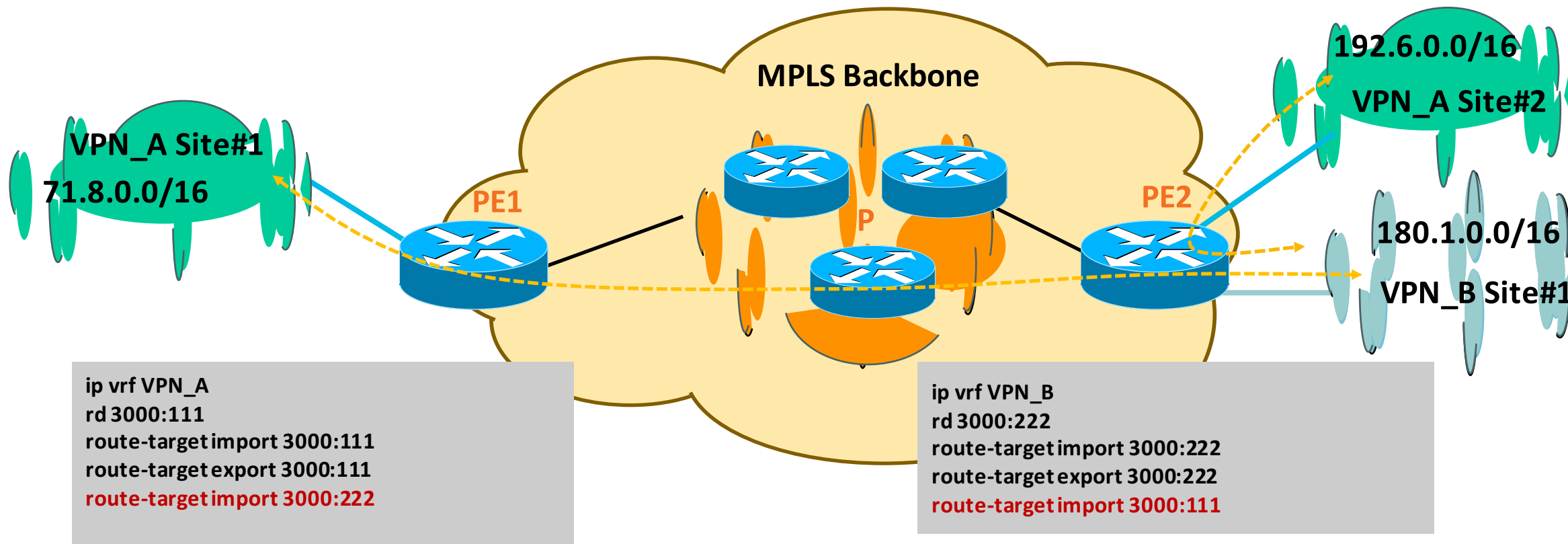
# MPLS-VPN Services

## 3. Extranet VPN

- MPLS based IP/VPN, by default, isolates one VPN customer from another
  - Separate **virtual routing table** for each VPN customer
- **Communication between VPNs may be required i.e., extranet**
  - External intercompany communication (dealers with manufacturer, retailer with wholesale provider, etc.)
  - **Management VPN**, shared-service VPN, etc.
- Needs to share the **import and export route-target (RT)** values within the VRFs of extranets.
  - **Export-map or import-map may be used for advanced extranet.**

# MPLS-VPN Services

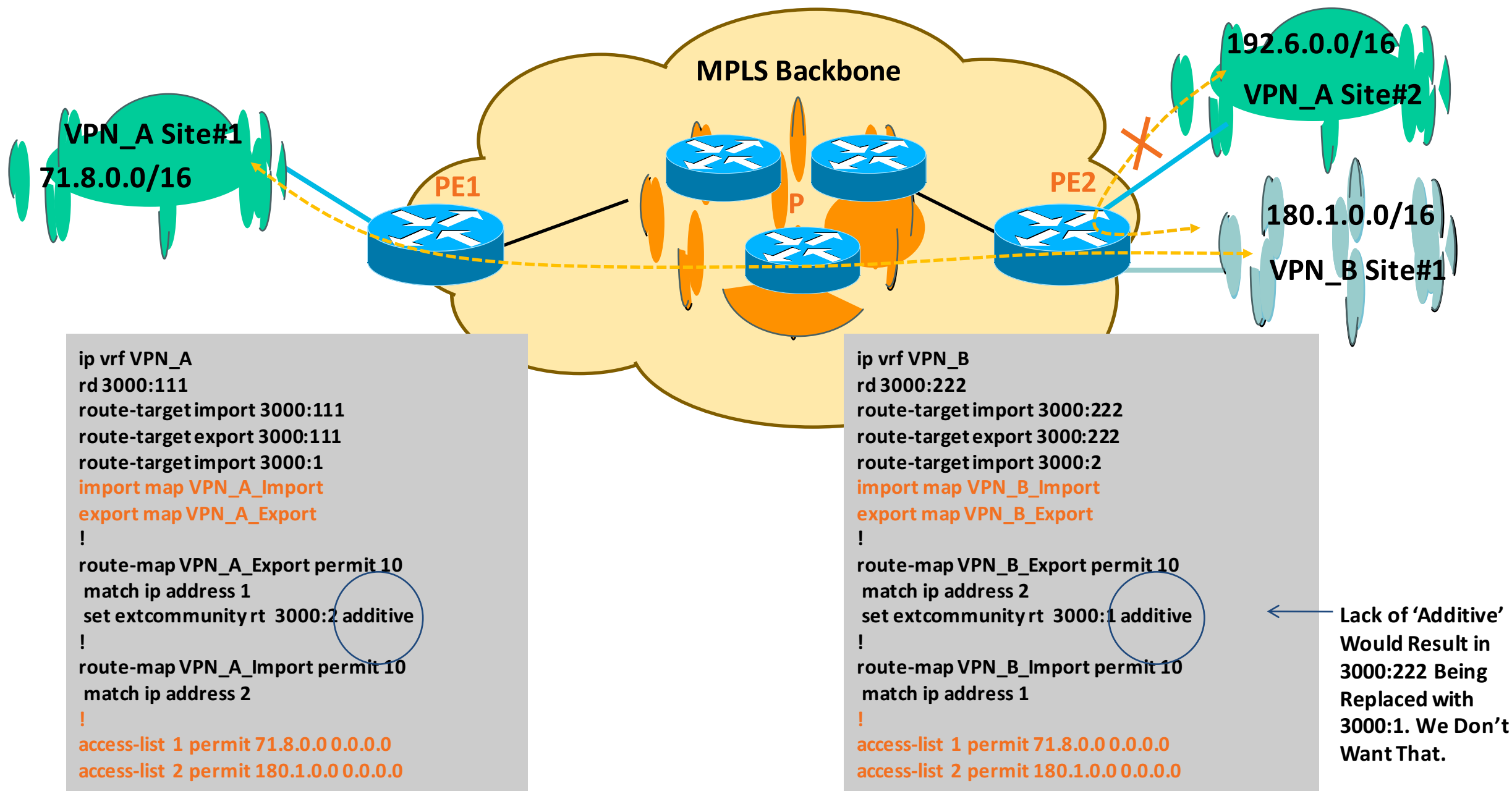
## 3. Extranet VPN – Simple Extranet (IOS Config sample)



**All Sites of Both VPN\_A and VPN\_B Can Communicate  
with Each Other**

# MPLS-VPN Services

## 3. Extranet VPN – Advanced Extranet (IOS Config sample)



**Only Site #1 of Both VPN\_A and VPN\_B Would Communicate with Each Other**

# Agenda

- IP/VPN Overview
- IP/VPN Services
  1. Load-Sharing for Multihomed VPN Sites
  2. Hub and Spoke Service
  3. Extranet Service
  4. **Internet Access Service**
  5. IP/VPN over IP Transport
  6. IPv6 VPN Service
- Best Practices
- Conclusion

# MPLS-VPN Services

## 4. Internet Access Service to VPN Customers

- Internet access service could be provided as another value-added service to VPN customers
- Security mechanism **must** be in place at both provider network and customer network
  - To protect from the Internet vulnerabilities
- **VPN customers benefit from the single point of contact for both Intranet and Internet connectivity**



# MPLS-VPN Services

## 4. Internet Access: Design Options

Four Options to Provide the Internet Service -

1. VRF specific default route with “global” keyword
2. Separate PE-CE sub-interface (non-VRF)
3. Extranet with Internet-VRF
4. VRF-aware NAT

# MPLS-VPN Services

## 4. Internet Access: Design Options

### 1. VRF specific default route

- 1.1 Static default route to move traffic from VRF to Internet (global routing table)
- 1.2 Static routes for VPN customers to move traffic from Internet (global routing table) to VRF

### 2. Separate PE-CE subinterface (non-VRF)

- May run BGP to propagate Internet routes between PE and CE

### 3. Extranet with Internet-VRF

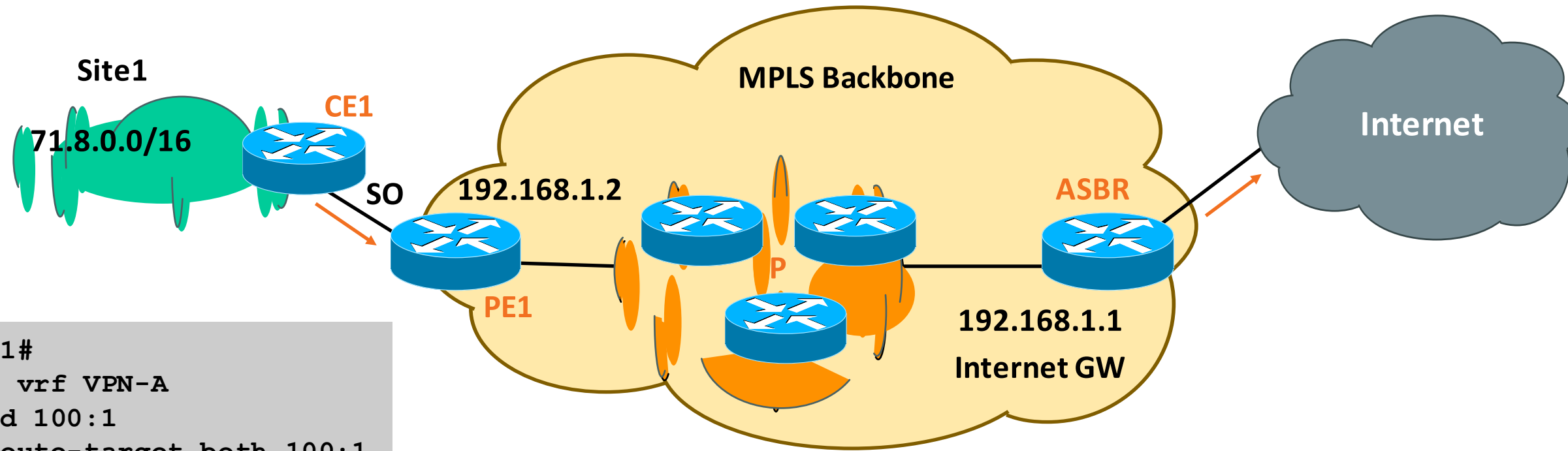
- VPN packets never leave VRF context; issue with overlapping VPN address

### 4. Extranet with Internet-VRF along with VRF-aware NAT

- VPN packets never leave VRF context; works well with overlapping VPN address

# IP/VPN Services: Internet Access

## 4.1 Option#1: VRF Specific Default Route



```
PE1#
ip vrf VPN-A
rd 100:1
route-target both 100:1
```

```
Interface Serial0
ip address 192.168.10.1 255.255.255.0
ip vrf forwarding VPN-A
```

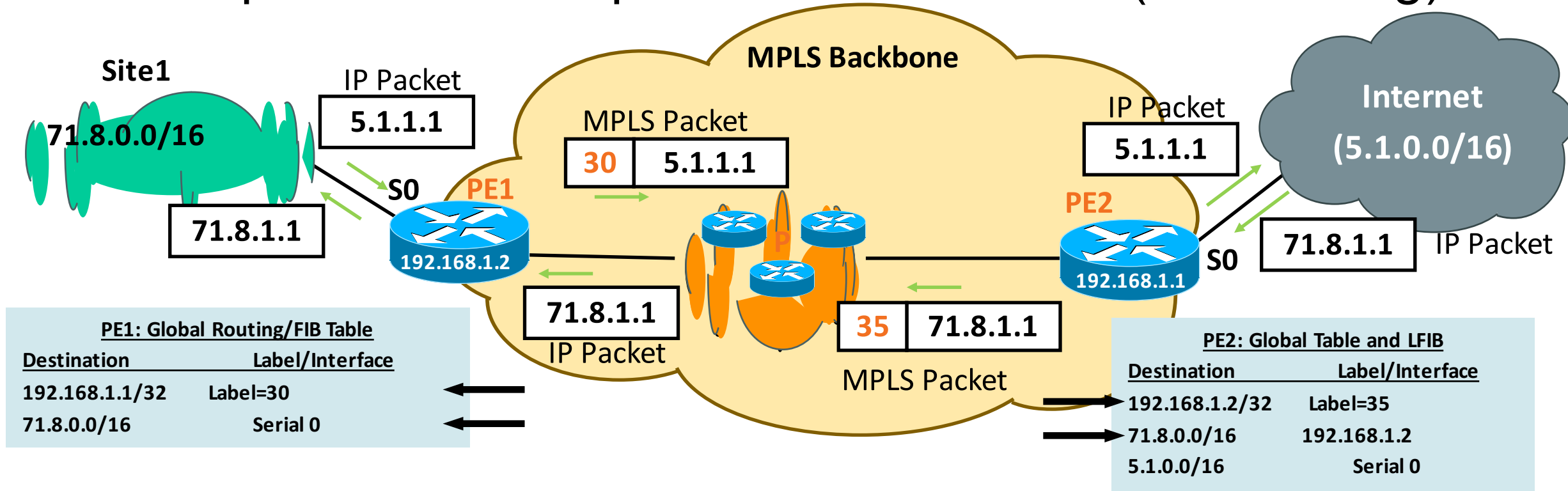
```
Router bgp 100
no bgp default ipv4-unicast
redistribute static
neighbor 192.168.1.1 remote 100
neighbor 192.168.1.1 activate
neighbor 192.168.1.1 next-hop-self
neighbor 192.168.1.1 update-source loopback0
```

```
ip route vrf VPN-A 0.0.0.0 0.0.0.0 192.168.1.1 global
ip route 71.8.0.0 255.255.0.0 Serial0
```

- A default route, pointing to the ASBR, is installed into the site VRF at **each PE**
- The static route, pointing to the VRF interface, is installed in the global routing table and redistributed into BGP

# IP/VPN Services: Internet Access

## 4.1 Option#1: VRF Specific Default Route (Forwarding)



### Pros

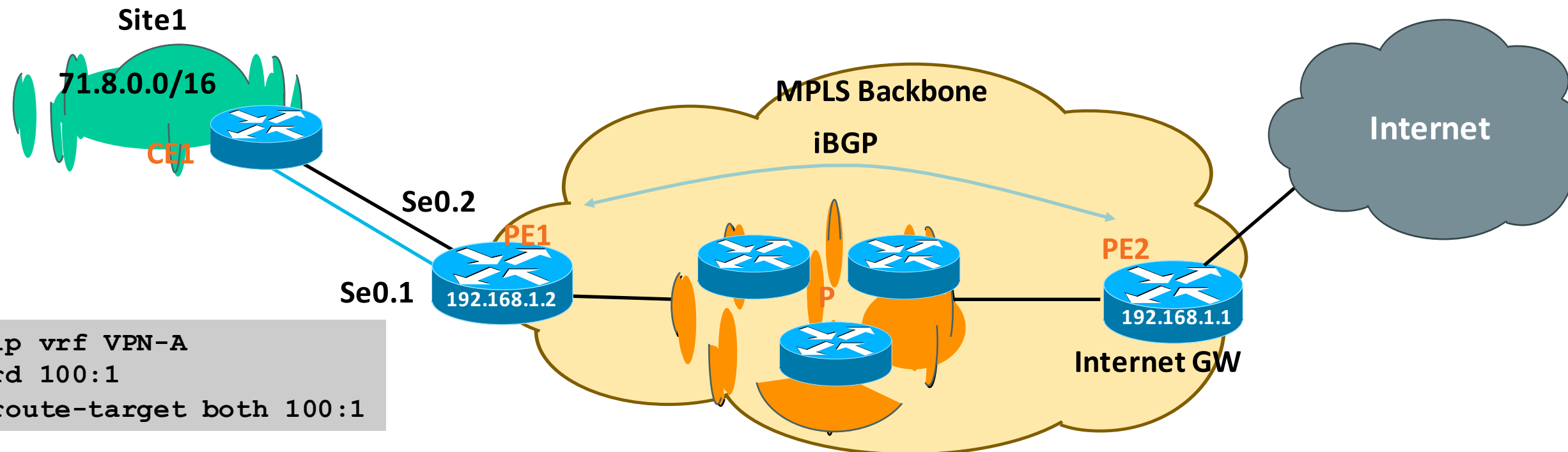
- Different Internet gateways
- Can be used for different VRFs
- PE routers need not to hold the Internet table
- Simple configuration

### Cons

- Using default route for Internet
- Routing does **not** allow any other default route for intra-VPN routing  
Increasing size of global routing table by leaking VPN routes
- Static configuration (possibility of traffic blackholing)

# IP/VPN Services: Internet Access

## 4.2 Option#2: Separate PE-CE Subinterfaces



```
ip vrf VPN-A
rd 100:1
route-target both 100:1
```

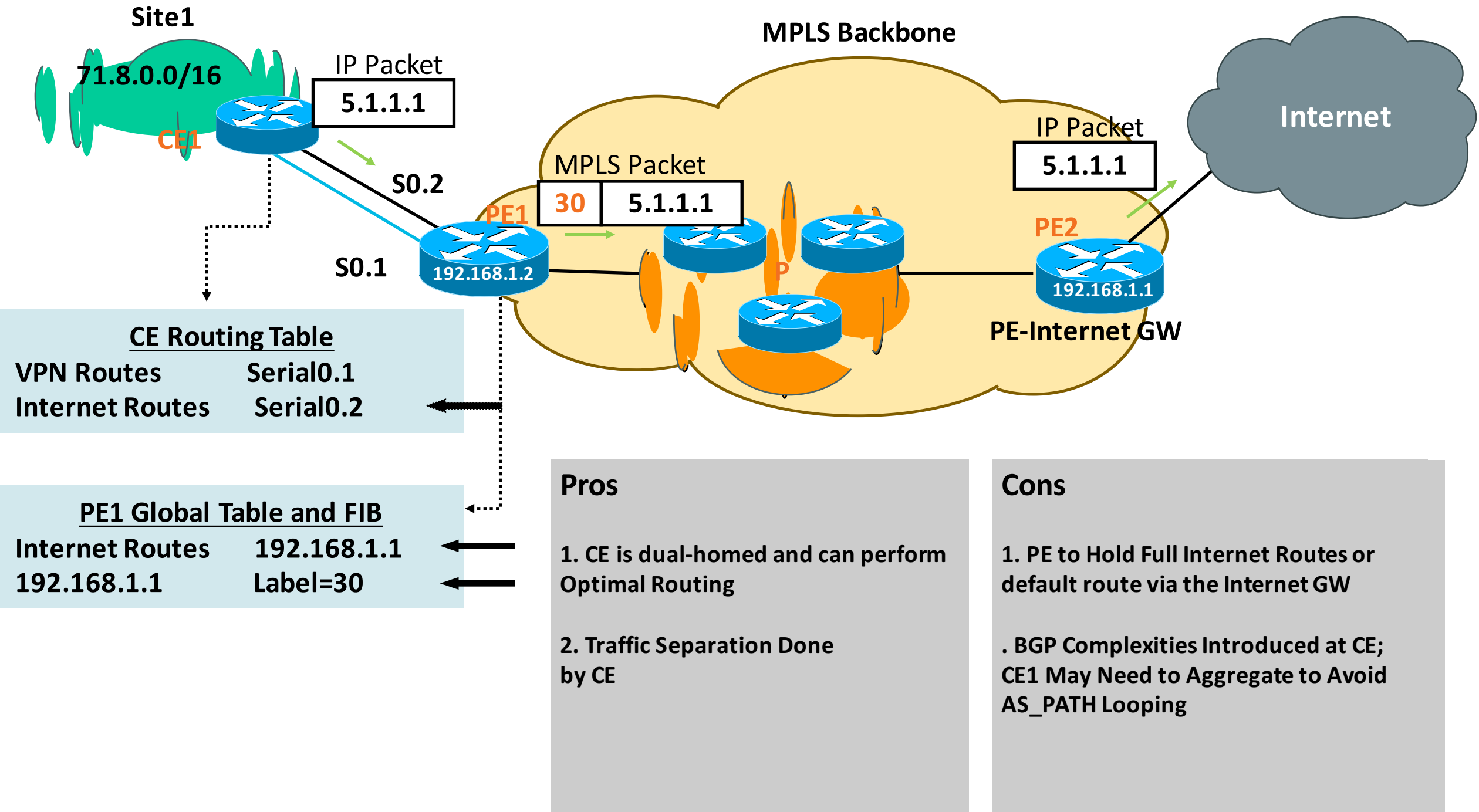
```
Interface Serial0.1
ip vrf forwarding VPN-A
ip address 192.168.20.1 255.255.255.0
frame-relay interface-dlci 100
!
Interface Serial0.2
ip address 71.8.10.1 255.255.0.0
frame-relay interface-dlci 200
!
```

```
Router bgp 100
no bgp default ipv4-unicast
neighbor 71.8.10.2 remote-as 502
```

- PE1-CE1 has one sub-interface associated to a VRF for VPN routing
- PE1-CE has another subinterface (global) for Internet routing
- PE1 may have eBGP peering with CE1 over the global interface and advertise full Internet routes or a default route to CE1
- PE2 must advertise VPN/site1 routes to the Internet.

# IP/VPN Services: Internet Access

## 4.2 Option#2: Separate PE-CE Subinterfaces (Forwarding)



# IP/VPN Services: Internet Access

## 4.3 Option#3: Extranet with Internet-VRF

- The Internet routes could be placed within the VRF at the Internet-GW i.e., ASBR
- VRFs for customers could 'extranet' with the Internet VRF and receive either *default, partial* or full Internet routes
  - Default route is recommended
- **Be careful** if multiple customer VRFs, at the same PE, are importing **full Internet routes**
- Works well **only** if the VPN customers don't have overlapping addresses


# IP/VPN Services: Internet Access

## 4.4 Option#4: Using VRF-Aware NAT

- If the VPN customers need Internet access without Internet routes, then **VRF-aware NAT** can be used at the Internet-GW i.e., ASBR
- The Internet GW doesn't need to have Internet routes either
- **Overlapping VPN addresses is no longer a problem**
- More in the “VRF-aware NAT” slides...



# Agenda

- IP/VPN Overview
- IP/VPN Services
  1. Load-Sharing for Multihomed VPN Sites
  2. Hub and Spoke Service
  3. Extranet Service
  4. Internet Access Service
  -  5. IP/VPN over IP Transport
  6. IPv6 VPN Service
- Best Practices
- Conclusion

# IP/VPN Services:

## 10. Providing MPLS/VPN over IP Transport

- MPLS/VPN (rfc2547) can also be deployed using IP transport
  - No MPLS needed in the core
- PE-to-PE IP tunnel is used, instead of MPLS tunnel, for sending MPLS/VPN packets
  - MPLS labels are still allocated for VPN prefixes by PE routers and used only by the PE routers
  - MPLS/VPN packet is encapsulated inside an IP header
- IP tunnel could be GRE, mGRE etc.

# IP/VPN Services:

## 10. Providing MPLS/VPN over IP Transport



- GRE/IP header and VPN label imposed on VPN traffic by PE1
- VPN traffic is forwarded towards egress PE using IP forwarding
- Egress PE2 decapsulates, and uses VPN label to forward packet to CE2

IP Packet

# Agenda

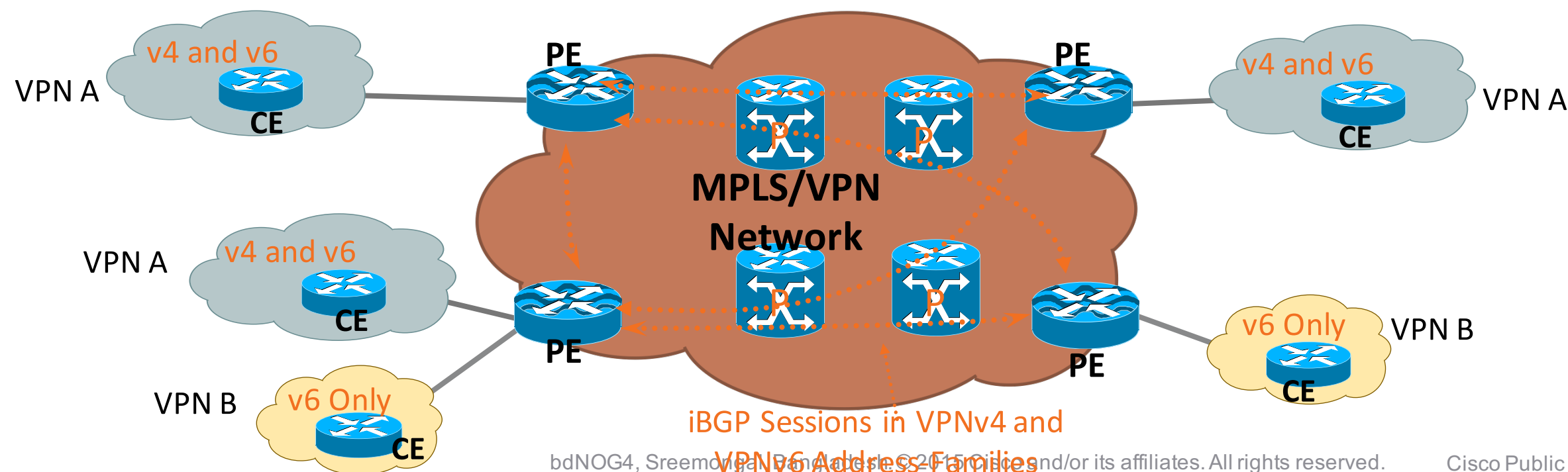
- IP/VPN Overview
- IP/VPN Services
  1. Load-Sharing for Multihomed VPN Sites
  2. Hub and Spoke Service
  3. Extranet Service
  4. Internet Access Service
  5. IP/VPN over IP Transport
  6. **IPv6 VPN Service**
- Best Practices
- Conclusion



# IP/VPN Services:

## 11. IPv6 VPN Service

- Similar to IPv4 VPN, IPv6 VPN can also be offered.
  - Referred to as “IPv6 VPN Provider Edge (6VPE)”.
- No modification on the MPLS core
  - Core can stay on IPv4
- PE-CE interface can be single-stack IPv6 or dual-stack
  - IPv4 and IPv6 VPNs can be offered on the same PE-CE interface
- Config and operation of IPv6 VPN are similar to IPv4 VPN



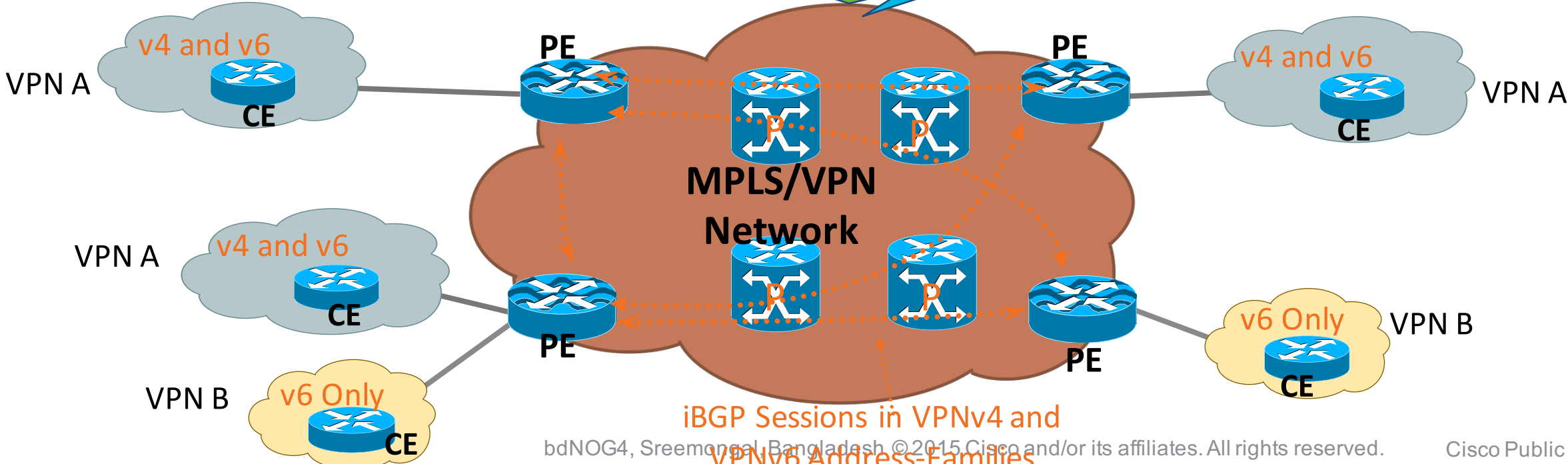
# IP/VPN Services:

## 11. IPv6 VPN Service

```
IOS_PE#
!
vrf definition v2
 rd 2:2
!
 address-family ipv6
  route-target export 2:2
  route-target import 2:2
!
router bgp 1
!
 address-family vpnv6
  neighbor 10.13.1.21 activate
  neighbor 10.13.1.21 send-community both
!
 address-family ipv6 vrf v2
  neighbor 200::2 remote-as 30000
  neighbor 200::2 activate
!
```

```
IOS-XR_PE#
!
vrf v2
!
 address-family ipv6 unicast
  route-target export 2:2
  route-target import 2:2
!
router bgp 1
 address-family vpnv6 unicast
!
 neighbor 10.13.1.21
  remote-as 30000
 address-family vpnv6 unicast
!
vrf v2
 rd 2:2
 address-family ipv6 unicast
!
 neighbor 200::2
  remote-as 30000
 address-family ipv6 unicast
!
```

```
NXOS_PE#
!
vrf context v2
 rd 2:2
!
 address-family ipv6 unicast
  route-target export 2:2
  route-target import 2:2
!
router bgp 1
 neighbor 10.13.1.21
  remote-as 1
 update-source loopback0
 address-family vpnv6 unicast
  send-community extended
!
vrf vpn1
 neighbor 200::2
  remote-as 30000
 address-family ipv6 unicast
!
```



# Agenda

- IP/VPN Overview
- IP/VPN Services
- **Best Practices**
- Conclusion

# Best Practices (1)

1. **Use RR to scale BGP**; deploy RRs in pair for the redundancy  
Keep RRs out of the forwarding paths and disable CEF (saves memory)
2. **Choose AS/IP format for RT and RD** i.e., ASN: X  
Reserve first few 100s of X for the internal purposes such as filtering
3. Consider **unique RD per VRF per PE**,  
Helpful for many scenarios such as multi-homing, hub&spoke etc.
4. **Don't use customer names** (V458:GodFatherNYC32ndSt) **as the VRF names**; nightmare for the NOC.  
Consider v101, v102, v201, v202, etc. and Use VRF description for naming
5. **Utilize SP's public address space for PE-CE IP addressing**  
Helps to avoid overlapping; Use **/31 subnetting** on PE-CE interfaces



# Best Practices (2)

## 6. Limit number of prefixes per-VRF and/or per-neighbor on PE

Max-prefix within VRF configuration; Suppress the inactive routes

Max-prefix per neighbor (PE-CE) within OSPF/RIP/BGP VRF af

## 7. Leverage BGP Prefix Independent Convergence (PIC) for fast convergence <100ms (IPv4 and IPv6):

- PIC Core
- PIC Edge
- Best-external advertisement
- Next-hop tracking (ON by default)

## 8. Consider RT-constraint for Route-reflector scalability

## 9. Consider 'BGP slow peer' for PE or RR – faster BGP convergence

# Agenda

- IP/VPN Overview
- IP/VPN Services
- Best Practices
- Conclusion



# Conclusion

- MPLS based IP/VPN is the most optimal L3VPN technology
  - Any-to-any IPv4 or IPv6 VPN topology
  - Partial-mesh, Hub and Spoke topologies also possible
- Various IP/VPN services for additional value/revenue
- IP/VPN paves the way for virtualization & Cloud Services
  - Benefits whether SP or Enterprise.

BUILT FOR  
THE HUMAN  
NETWORK



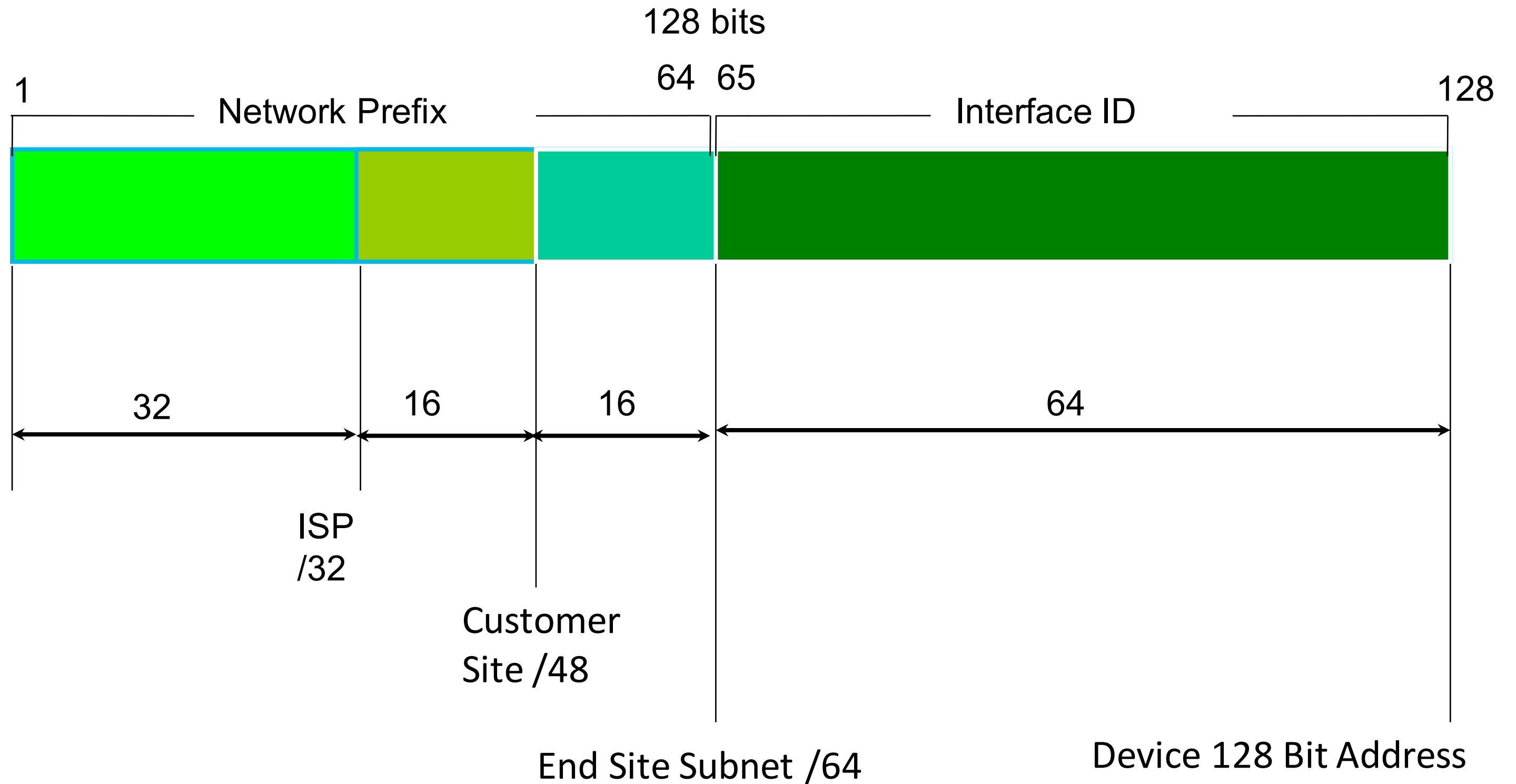
# IPv6 Addressing

- An IPv6 address is 128 bits long
- So the number of addresses are  $2^{128}$   
=340282366920938463463374607431768211455  
(39 decimal digits)  
=0xffffffffffffffffffffffffffffffff (32 hexadecimal digits)
- In hex 4 bit (nibble) is represented by a hex digit
- So 128 bit is reduced down to 32 hex digit

# IPv6 Address Representation

- Hexadecimal values of eight 16 bit fields
  - X:X:X:X:X:X:X:X (X=16 bit number, ex: A2FE)
  - 16 bit number is converted to a 4 digit hexadecimal number
- Example:
  - FE38:DCE3:124C:C1A2:BA03:6735:EF1C:683D
  - Abbreviated form of address
    - 4EED:0023:0000:0000:0000:036E:1250:2B00
    - →4EED:23:0:0:0:36E:1250:2B00
    - →4EED:23::36E:1250:2B00
    - (Null value can be used only once)

# IPv6 addressing structure



# IPv6 addressing model

- **IPv6 Address type**

- Unicast

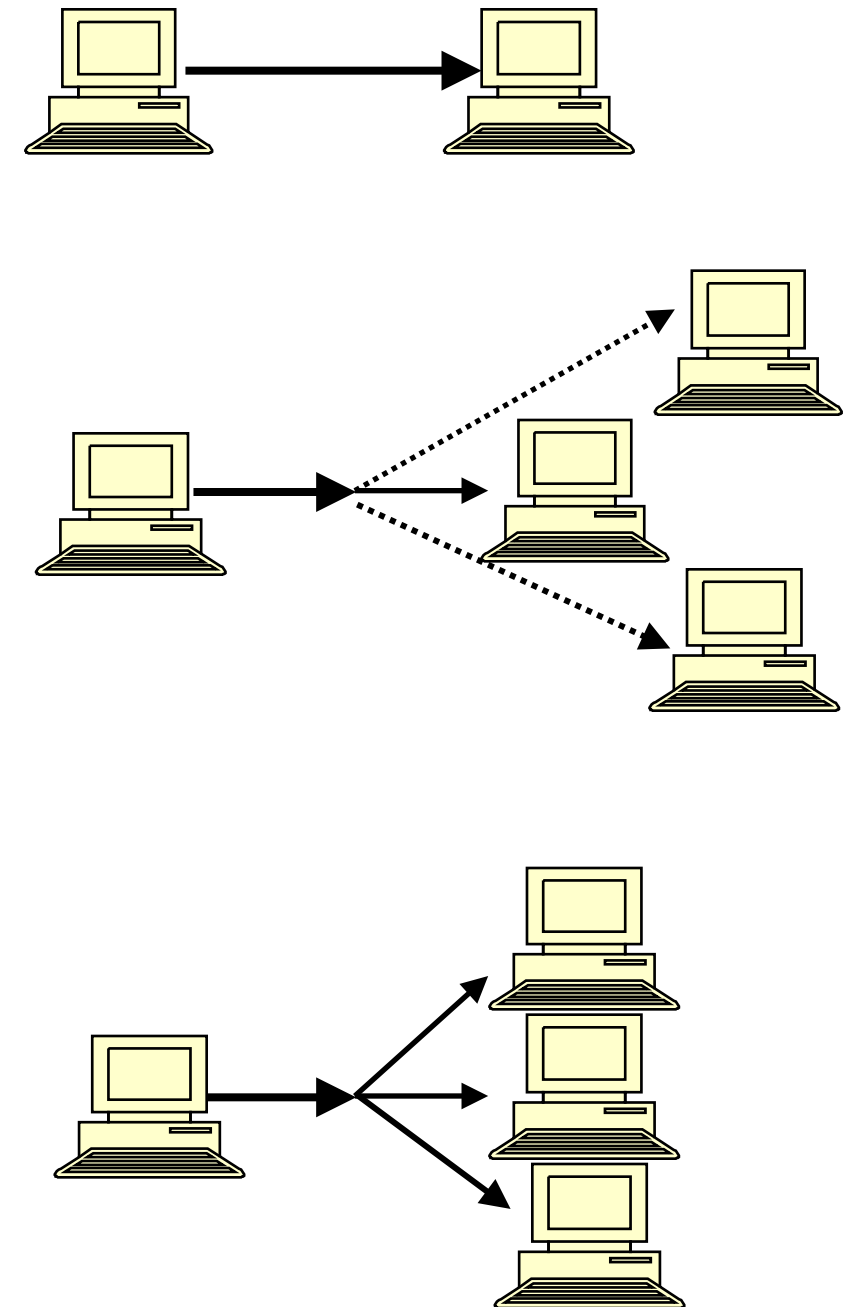
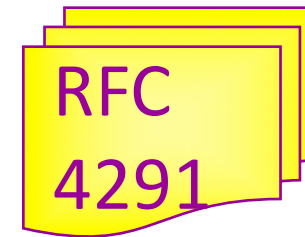
An identifier for a single interface

- Anycast

An identifier for a set of interfaces

- Multicast

An identifier for a group of nodes





# Addresses Without a Network Prefix

- Localhost `::1/128`
- Unspecified Address `::/128`
- IPv4-mapped IPv6 address `::ffff/96 [a.b.c.d]`
- IPv4-compatible IPv6 address `::/96 [a.b.c.d]`

# Local Addresses With Network Prefix

- Link Local Address
  - A special address used to communicate within the local link of an interface
  - i.e. anyone on the link as host or router
  - This address in packet destination that packet would never pass through a router
  - fe80::/10

# Local Addresses With Network Prefix

- Unique Local IPv6 Unicast Address
  - Addresses similar to the RFC 1918 / private address like in IPv4 but will ensure uniqueness
  - A part of the prefix (40 bits) are generated using a pseudo-random algorithm and it's improbable that two generated ones are equal
  - fc00::/7
  - Example webtools to generate ULA prefix
    - <http://www.sixxs.net/tools/grh/ula/>
    - <http://www.goebel-consult.de/ipv6/createLULA>

# Global Addresses With Network Prefix

- IPv6 Global Unicast Address

- Global Unicast Range: 0010 2000::/3

- 0011 3000::/3

- All five RIRs are given a /12 from the /3 to further distribute within the RIR region

- APNIC 2400:0000::/12

- ARIN 2600:0000::/12

- AfriNIC 2C00:0000::/12

- LACNIC 2800:0000::/12

- Ripe NCC 2A00:0000::/12

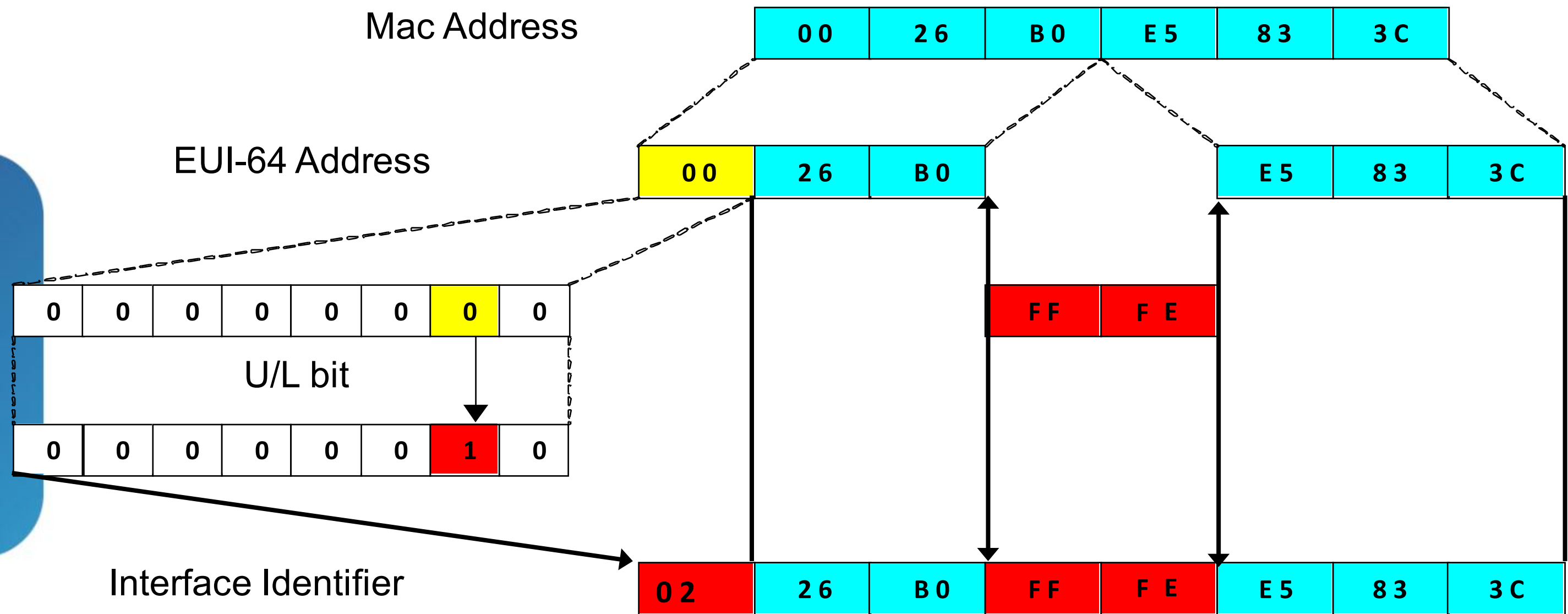
# Examples and Documentation Prefix

- Two address ranges are reserved for examples and documentation purpose by RFC 3849
  - For example 3fff:ffff::/32
  - For documentation 2001:0DB8::/32

# Interface ID

- The lowest-order 64-bit field addresses may be assigned in several different ways:
  - auto-configured from a 48-bit MAC address expanded into a 64-bit EUI-64
  - assigned via DHCP
  - manually configured
  - auto-generated pseudo-random number
  - possibly other methods in the future

# EUI-64



# IPv6 Neighbor Discovery (ND)

- IPv6 use multicast (L2) instead of broadcast to find out target host MAC address
- It increases network efficiency by eliminating broadcast from L2 network
- IPv6 ND use ICMP6 as transport
  - Compared to IPv4 ARP no need to write different ARP for different L2 protocol i.e. Ethernet etc.



# IPv6 Neighbor Discovery (ND)

- Solicited Node Multicast Address
  - Start with FF02:0:0:0:0:1:ff::/104
  - Last 24 bit from the interface IPV6 address
- Example Solicited Node Multicast Address
  - IPV6 Address 2406:6400:0:0:0:0:0000:0010
  - Solicited Node Multicast Address is FF02:0:0:0:0:1:ff00:0010
- All host listen to its solicited node multicast address corresponding to its unicast and anycast address (If defined)

# IPv6 Neighbor Discovery (ND)

- Host A would like to communicate with Host B
- Host A IPv6 global address 2406:6400::10
- Host A IPv6 link local address fe80::226:bbff:fe06:ff81
- Host A MAC address 00:26:bb:06:ff:81
- Host B IPv6 global address 2406:6400::20
- Host B Link local UNKNOWN [Gateway if outside the link]
- Host B MAC address UNKNOWN
- How Host A will create L2 frame for Host B?

# IPv6 Neighbor Discovery (ND)

## Host A

IPv6 global address: 2406:6400::0010

IPv6 Link local: fe80::0226:bbff:fe06:ff81

MAC address: 00:26:bb:06:ff:81

Listen to other then above:

FF02::1

[All node multicast]

FF02:0:0:0:0:1:ff00:0010 [Solicited node m.cast unicast]

FF02:0:0:0:0:1:ff06:ff81 [Solicited node m.cast link local]

Packet

S: 2406:6400::0010 D:2406:6400::0020

ICMP6 NS Type 135

S: fe80::0226:bbff:fe06:ff81  
D:FF02:0:0:0:0:1:ff00:0020

Frame

S: 00:26:bb:06:ff:81 D 33:33:ff:00:00:20  
Ethernet reserved IPv6 m.cast: 33:33:xx:xx:xx:xx



Multicast enable switch: Unicast by IGMP snooping  
Non multicast enable switch: broadcast, PC LAN card filter or discard

## Host B

IPv6 global address: 2406:6400::0020

IPv6 Link local: fe80::0226:bbff:fe06:ff82 [Unknown to A]

MAC address: 00:26:bb:06:ff:82 [Unknown to A]

Listen to other then above:

FF02::1

[All node multicast]

FF02:0:0:0:0:1:ff00:0020 [Solicited node m.cast unicast]

FF02:0:0:0:0:1:ff06:ff82 [Solicited node m.cast link local]

Packet

S: 2406:6400::0020 D:2406:6400::0010

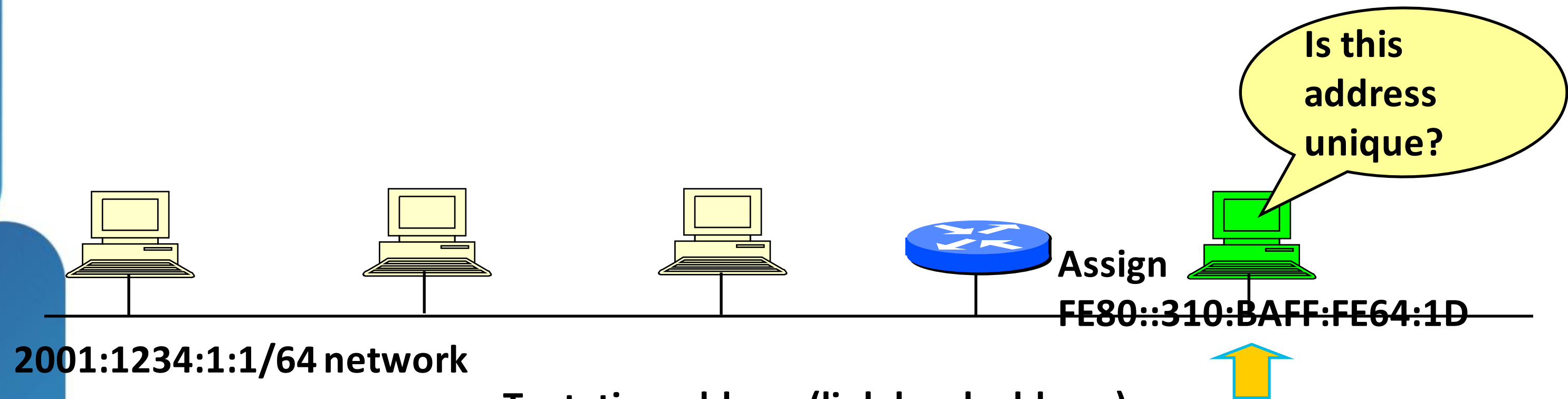
ICMP6 NA Type 136

S: fe80::0226:bbff:fe06:ff82  
D:fe80::0226:bbff:fe06:ff81

Frame

S: 00:26:bb:06:ff:82 D 00:26:bb:06:ff:81

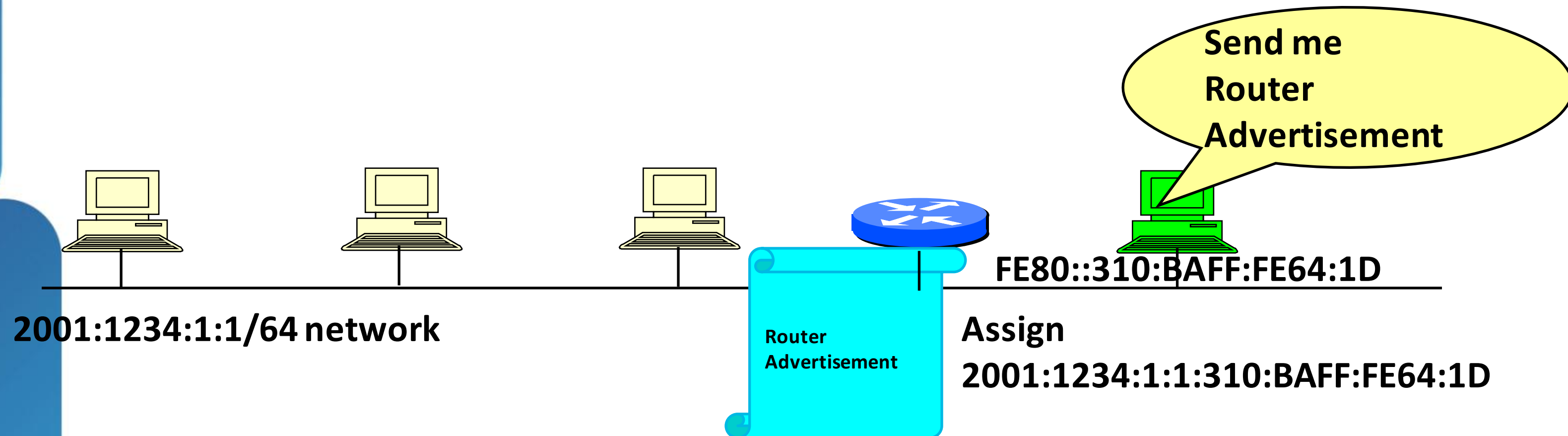
# IPv6 autoconfiguration



Tentative address (link-local address)  
Well-known link local prefix +Interface ID (EUI-64)  
Ex: FE80::310:BAFF:FE64:1D

1. A new host is turned on.
2. Tentative address will be assigned to the new host.
3. Duplicate Address Detection (DAD) is performed. First the host transmit a Neighbor Solicitation (NS) message to the solicited node multicast address (FF02::1:FF64:001D) corresponding to its to be used address
5. If no Neighbor Advertisement (NA) message comes back then the address is unique.
6. FE80::310:BAFF:FE64:1D will be assigned to the new host.

# IPv6 autoconfiguration



1. The new host will send Router Solicitation (RS) request to the all-routers multicast group (FF02::2).
2. The router will reply Routing Advertisement (RA).
3. The new host will learn the network prefix. E.g, 2001:1234:1:1/64
4. The new host will assigned a new address Network prefix+Interface ID E.g,  
2001:1234:1:1:310:BAFF:FE64:1D

BUILT FOR  
THE HUMAN  
NETWORK

