MPLS Workshop

Jan 15th to Jan 19th 2009

Abdul Rahim , arahim@cisco.com

Configuring MPLS

Configuring MPLS

Mandatory:

- Enable CEF switching.
- Configure Tag Distribution Protocol or Label Distribution Protocol on every label-enabled interface.

Optional:

- Configure MTU size for labeled packets.
- Configure IP TTL propagation.
- Configure conditional label advertising.

Configuring LDP

Global

```
ip cef <distributed>
mpls label protocol <ldp | tdp | both>
tag-switching tdp router-id Loopback0
mpls ldp explicit-null (optional)
no mpls ip propagate-ttl (optional)
```

Interface

```
mpls ip  or  tag-switching ip (enables this interface for MPLS forwarding)
mpls label protocol ldp
```

(optional, if you want to run LDP on this interface only, while other interfaces don't run LDP or run another label protocol such as TDP)

Configuring Conditional Label Distribution

Router(config)#

```
tag-switching advertise-tags for net-acl [ to tdp-acl ]
```

- By default, labels for all destinations are announced to all LDP/TDP neighbors.
- This command enables you to selectively advertise some labels to some LDP/TDP neighbors.
- Conditional label advertisement only works over frame-mode interfaces.
- Parameters:
  - Net-ACL – the IP ACL that selects the destinations for which the labels will be generated.
  - TDP-ACL – the IP ACL that selects the TDP neighbors that will receive the labels.

Conditional Label Distribution Example

- The customer is already running IP infrastructure.
- MPLS is only needed to support MPLS/VPN services.
  - Labels should only be generated for loopback interfaces (BGP next-hops) of all routers.
  - All loopback interfaces are in one contiguous address block (192.168.254.0/24).

Conditional Label Distribution Router Configuration

- Enable conditional label advertisment

```
no tag-switching advertise-tags
!
! Configure conditional advertisments
!
tag-switching advertise-tags for 90 to 91
!
access-list 90 permit ip 192.168.254.0 0.0.0.255
access-list 91 permit ip any
```

Monitoring LDP

- `show mpls interface <x> detail`
- `show mpls ldp discovery`
- `show mpls ldp neighbor`
- `show mpls ip/ldp binding <prefix> <prefix-length>`
- `show mpls forwarding-table <prefix> <prefix-length>`
- `sh ip cef <prefix>`
- `show mpls ldp parameters`

Show mpls interface

```
mpls-7200a#sh mpls interface
Interface              IP         Tunnel   Operational
Ethernet3/0            Yes (ldp)  No       Yes
```

```
mpls-7200a#sh mpls interface ethernet3/0 detail
Interface Ethernet3/0:
        IP labeling enabled (ldp)
        ……..<snip>…….
          Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
          MTU = 1500
```

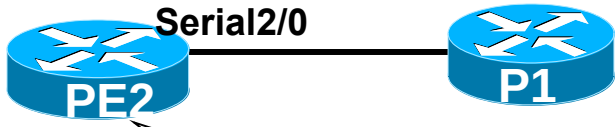
Show mpls interface (contd..)

- "sh mpls interface [detail]"

  Lists whether MPLS is enabled and the application that enabled MPLS on the interface

```
PE2#sh mpls interface
Interface              IP            Tunnel   Operational
Serial2/0              Yes (ldp)     No       Yes
PE2#
```

```
PE2#sh mpls interface ser2/0 detail
Interface Serial2/0:
        IP labeling enabled (ldp)
        LSP Tunnel labeling not enabled
        BGP tagging not enabled
        Tagging operational
        Fast Switching Vectors:
          IP to MPLS Fast Switching Vector
          MPLS Turbo Vector
        MTU = 1508
PE2#
```



MPLS Enabled

LDP Enabled

MPLS MTU

```
!
interface Serial2/0
 description To P1 ser2/0
 ip address 10.13.2.6/30
 mpls label protocol ldp
 tag-switching ip
 tag-switching mtu 1508
!
```

Show mpls interface (contd..)

- This slide is to show that BGPipv4+label (or MP-eBGP) is another application that can enable MPLS; *what's different here -*

```
RSP-PE-SOUTH-6#sh mpls int
Interface              IP            Tunnel   Operational
Fddi1/0/0              Yes (ldp)     No       Yes
ATM1/1/0.108           No            No       Yes
RSP-PE-SOUTH-6#
```

MPLS is Operational.

LDP not enabled

```
RSP-PE-SOUTH-6#sh mpls int ATM1/1/0.108 detail
Interface ATM1/1/0.108:
        IP labeling not enabled
        LSP Tunnel labeling not enabled
        BGP tagging enabled
        Tagging operational
        Optimum Switching Vectors:
          IP to MPLS Feature Vector
          MPLS Feature Vector
        Fast Switching Vectors:
          IP to MPLS Fast Feature Switching Vector
          MPLS Feature Vector
        MTU = 4470
RSP-PE-SOUTH-6#
```

LDP not enabled

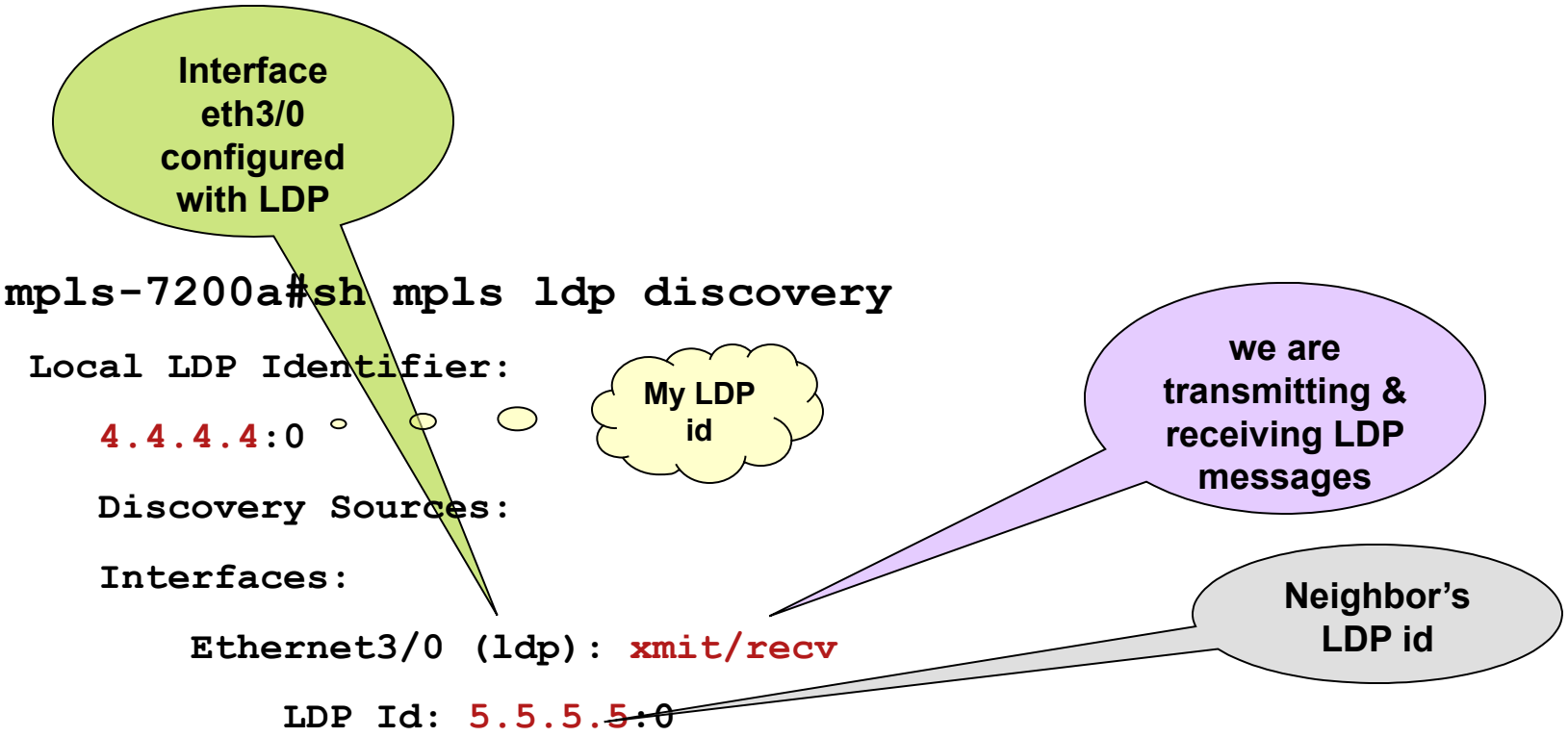
BGP+Label Enabled

MPLS MTU

LDP discovery/adjacency: commands and debugs

- `show mpls ldp discovery`
- `debug mpls ldp transport`
- `debug mpls ldp session io`

LDP discovery



- "debug mpls ldp transport events"

  Should give information regarding whether the HELLOS are advertised/received

LDP adjacency debugs

LDP discovery, connection setup and shutdown events

```
mpls-7200a#debug mpls ldp transport events
debugging for LDP discovery and connection setup / shutdown events
2d11h: ldp: Send ldp hello; Ethernet3/0, src/dst 10.0.3.4/224.0.0.2, inst_id 0
2d11h: ldp: Rcvd ldp hello; Ethernet3/0, from 10.0.3.5 (5.5.5.5:0), intf_id 0, opt
0xC
shutting neighbor
2d11h: %CLNS-5-ADJCHANGE: ISIS: Adjacency to mpls-12008a (Ethernet3/0) Down, hold
time expired
2d11h: ldp:Discovery hold timer expired for adj 0x17D45A0, 5.5.5.5:0,will close conn
2d11h: ldp: Discovery hold timer expired for adj 0x17D45A0; 5.5.5.5:0
2d11h: ldp:     adj_addr/adj_xport_addr: 10.0.3.5/5.5.5.5
2d11h: ldp: LDP ptcl SM; close xport request for adj 0x0
2d11h: ldp: Close LDP transport conn for adj 0x17D45A0
2d11h: ldp: Closing ldp conn 4.4.4.4:646 <-> 5.5.5.5:11012, adj 0x17D45A0
2d11h: ldp: Adj 0x17D45A0; state set to closed
2d11h: ldp: Send ldp hello; Ethernet3/0, src/dst 10.0.3.4/224.0.0.2, inst_id 0
```

LDP session i/o debug

LDP session I/O, excluding periodic Keep Alives

```
mpls-7200a#debug mpls ldp session io <all>
bringing neighbor down
2d11h: %CLNS-5-ADJCHANGE: ISIS: Adjacency to mpls-12008a (Ethernet3/0) Down, hold
time expired
2d11h: ldp: Sent notif msg to 5.5.5.5:0 (pp 0x17A0870)
.............
bringing neighbor up
2d11h: %CLNS-5-ADJCHANGE: ISIS: Adjacency to mpls-12008a (Ethernet3/0) Up, new
adjacency
2d11h: ldp: Rcvd init msg from 5.5.5.5 (pp 0x0)
2d11h: ldp: Sent init msg to 5.5.5.5:0 (pp 0x0)
2d11h: ldp: Sent keepalive msg to 5.5.5.5:0 (pp 0x0)
2d11h: ldp: Rcvd keepalive msg from 5.5.5.5:0 (pp 0x0)
2d11h: ldp: Sent address msg to 5.5.5.5:0 (pp 0x186CB38)
2d11h: ldp: Sent label mapping msg to 5.5.5.5:0 (pp 0x186CB38)
.............
```

LDP neighbor

```
mpls-7200a#sh mpls ldp neighbor
    Peer LDP Ident: 5.5.5.5:0; Local LDP Ident 4.4.4.4:0
       TCP connection: 5.5.5.5.11000 - 4.4.4.4.646
       State: Oper; Msgs sent/rcvd: 268/264; Downstream Up time: 03:41:45
       LDP discovery sources:
          Ethernet3/0, Src IP addr: 10.0.3.5
       Addresses bound to peer LDP Ident:
          10.0.3.5        10.0.4.5        10.0.5.5        5.5.5.5
```

LDP neighbor (contd..)

- LDP session is a TCP session (port = 646)
- Multiple links between two routers still mean single LDP session.

```
PE1#sh mpls ldp neighbor
    Peer LDP Ident: 10.13.1.101:0; Local LDP Ident 10.13.1.61:0
        TCP connection: 10.13.1.101.11031 - 10.13.1.61.646
        State: Oper; Msgs sent/rcvd: 58/60; Downstream
        Up time: 00:39:27
        LDP discovery sources:
          Ethernet0/0, Src IP addr: 10.13.1.5
          Ethernet1/0, Src IP addr: 10.13.1.9
        Addresses bound to peer LDP Ident:
          10.13.1.9       10.13.1.5       10.13.2.5     10.13.1.101

PE1#
PE1#sh tcp brief| i 646
43ABB020  10.13.1.101.11031        10.13.1.61.646         ESTAB
PE1#
```

LDP_ID

Unsolicited Label Distribution*

Interfaces on which peer is discovered

Peer's Connected int

LDP binding commands

- “sh mpls ip binding detail”

  Lists all prefixes with labels & LDP neighbors

- “sh mpls ip binding <*prefix*> <mask> det”

  Lists ACLs (if any), *prefix* bindings, and LDP neighbors. Notice “Advertised to:” field.

- “sh mpls ip binding advertisement-acls”

  Lists LDP filter, if there is any, on the first line. Prefixes followed by “Advert acl(s):” are advertised via LDP, others are not.

LIB information

```
mpls-7200a#sh mpls ip binding 12.12.12.12 32
  12.12.12.12/32
        in label:     21
        out label:    19         lsr: 5.5.5.5:0         in use
```

```
mpls-7200a#sh mpls ldp binding 12.12.12.12 32
  tib entry: 12.12.12.12/32, rev 48
        local binding:  tag: 21
        remote binding: tsr: 5.5.5.5:0, tag: 19
```

LDP binding related debugs

```
mpls-7200a#debug mpls ldp bindings

shutting neighbor

2d11h: %CLNS-5-ADJCHANGE: ISIS: Adjacency to mpls-12008a (Ethernet3/0) Down, hold
time expired

2d11h: tagcon: tibent(5.5.5.5/32): label imp-null from 5.5.5.5:0 removed

2d11h: tagcon: route_tag_change for: 5.5.5.5/32

        inlabel 16, outlabel withdrwn, nexthop lsr 5.5.5.5:0, reason response to
find_route_tags

2d11h: tagcon: Deassign peer id; 5.5.5.5:0: id 0

2d11h: tagcon: tc_iprouting_table_change: 5.5.5.5/255.255.255.255, event 0x2

2d11h: tagcon: rib change: 5.5.5.5/255.255.255.255; event 0x2; ndb attrflags
0x1000000;

ndb->pdb_index/pdb->index 0x3/0x3

2d11h: tagcon: rib change: 5.5.5.5/255.255.255.255; event 0x2; ndb attrflags
0x1000000;

ndb->pdb_index/pdb->index 0x3/undef
```

LDP Advertisement related debugs

```
mpls-7200a#debug mpls ldp advertisements

shutting neighbor

2d11h: %CLNS-5-ADJCHANGE: ISIS: Adjacency to mpls-12008a (Ethernet3/0) Down,
hold time expired

2d11h: tagcon: Deassign peer id; 5.5.5.5:0: id 0

activating neighbor

2d11h: %CLNS-5-ADJCHANGE: ISIS: Adjacency to mpls-12008a (Ethernet3/0) Up,
new adjacency

2d11h: tagcon: Assign peer id; 5.5.5.5:0: id 0

2d11h: tagcon: peer 5.5.5.5:0 (pp 0x17AFAE0): advertise 4.4.4.4

2d11h: tagcon: Advertise labels: Clear LDP_CTX_TCB_FLAGS_ENULL_RECFG

2d11h: tagcon: peer 5.5.5.5:0 (pp 0x17AFAE0): advertise 4.4.4.4/32, label 3
(imp-null) (#32)
```

LFIB information

- `show mpls forwarding-table <prefix> <prefix-length>`
- `sh ip cef <prefix> internal`

Looking at LFIB

Looking at LFIB on 12008a

```
mpls-12008a#sh mpls forwarding 12.12.12.12 32 detail
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
19     19          12.12.12.12/32   498        Et2/0       10.0.4.11
       MAC/Encaps=14/18, MTU=1500, Tag Stack{19}
       AABBCC000502AABBCC0004028847 00013000
       No output feature configured
 Per-destination load-sharing, slots: 0 2 4 6 8 10 12 14
       19          12.12.12.12/32   498        Et3/0       10.0.5.11
       MAC/Encaps=14/18, MTU=1500, Tag Stack{19}
       AABBCC000503AABBCC0004038847 00013000
       No output feature configured
 Per-destination load-sharing, slots: 1 3 5 7 9 11 13 15
```

Ethertype= 8847 Label Value in MPLS shim= 13 Hex=19 dec

Destination MAC= AABBCC000502 Source MAC= AABBCC000402

CEF command

```
mpls-12008a#sh ip cef 12.12.12.12 internal

12.12.12.12/32, version 24, epoch 0, per-destination sharing
0 packets, 0 bytes
  tag information set, local tag: 19
  via 10.0.4.11, Ethernet2/0, 0 dependencies
    traffic share 1
    next hop 10.0.4.11, Ethernet2/0
    valid adjacency
    tag rewrite with Et2/0, 10.0.4.11, tags imposed: {19}
  via 10.0.5.11, Ethernet3/0, 0 dependencies
    traffic share 1
    next hop 10.0.5.11, Ethernet3/0
    valid adjacency
    tag rewrite with Et3/0, 10.0.5.11, tags imposed: {19}
0 packets, 0 bytes switched through the prefix
tmstats: external 0 packets, 0 bytes
         internal 0 packets, 0 bytes
  Load distribution: 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 (refcount 1)
```

Hash	OK	Interface	Address	Packets	Tags imposed
1	Y	Ethernet2/0	10.0.4.11	0	{19}
2	Y	Ethernet3/0	10.0.5.11	0	{19}
3	Y	Ethernet2/0	10.0.4.11	0	{19}
4	Y	Ethernet3/0	10.0.5.11	0	{19}
5	Y	Ethernet2/0	10.0.4.11	0	{19}
6	Y	Ethernet3/0	10.0.5.11	0	{19}
7	Y	Ethernet2/0	10.0.4.11	0	{19}
8	Y	Ethernet3/0	10.0.5.11	0	{19}
9	Y	Ethernet2/0	10.0.4.11	0	{19}
10	Y	Ethernet3/0	10.0.5.11	0	{19}
11	Y	Ethernet2/0	10.0.4.11	0	{19}
12	Y	Ethernet3/0	10.0.5.11	0	{19}
13	Y	Ethernet2/0	10.0.4.11	0	{19}
14	Y	Ethernet3/0	10.0.5.11	0	{19}
15	Y	Ethernet2/0	10.0.4.11	0	{19}
16	Y	Ethernet3/0	10.0.5.11	0	{19}

Monitoring LDP: LDP parameters

```
mpls-7200a#sh mpls ldp parameters
Protocol version: 1
Downstream label generic region: min label: 16; max label: 100000
Session hold time: 180 sec; keep alive interval: 60 sec
Discovery hello: holdtime: 15 sec; interval: 5 sec
Discovery targeted hello: holdtime: 180 sec; interval: 5 sec
Downstream on Demand max hop count: 255
TDP for targeted sessions
LDP initial/maximum backoff: 15/120 sec
LDP loop detection: off
```

Forwarding traffic down the LSP

```
mpls-7200a#sh mpls forwarding-table 12.12.12.12
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
21     19          12.12.12.12/32   0          Et3/0       10.0.3.5
Note: Bytes tag switched this will increment if packets are being tag
switched using this entry
```

```
mpls-12008a#sh mpls forwarding-table label 19
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
19     19          12.12.12.12/32   498        Et2/0       10.0.4.11
       19          12.12.12.12/32   1176       Et3/0       10.0.5.11
```

```
mpls-12008b#sh mpls forwarding-table labels 19
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
19     Pop tag     12.12.12.12/32   4176       Et1/0       10.0.17.12
```

LDP binding and advertisemsnt debugs

- Be Careful on the production routers
- “debug mpls ldp advertisements”

  Useful to see label bindings that are advertised
- “debug mpls ldp binding”

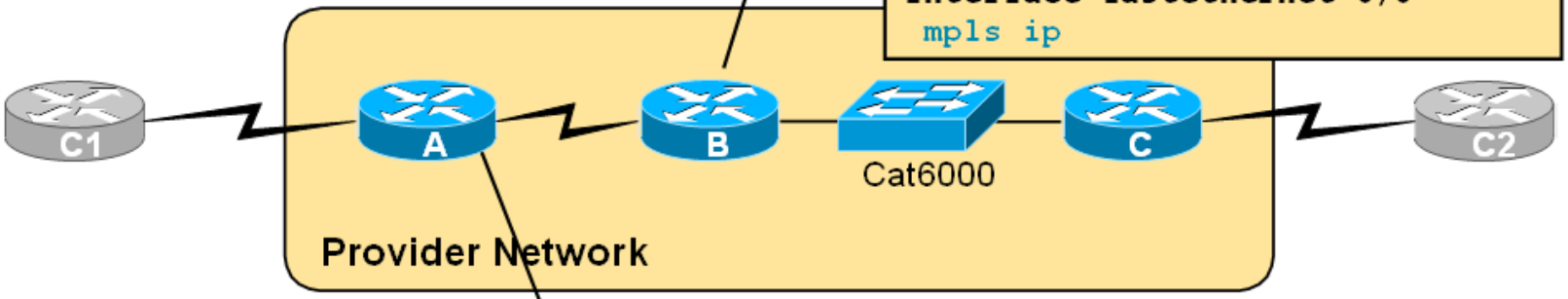
  Useful to see label bindings that are received
- “debug mpls ldp message sent|received”

  Useful for the protocol understanding purposes

MPLS Configuration Example

Enable MPLS on all core interfaces in your network.

```
ip cef
interface hssi 1/0
 mpls ip
interface fastethernet 0/0
 mpls ip
```



Never run MPLS with your customers.

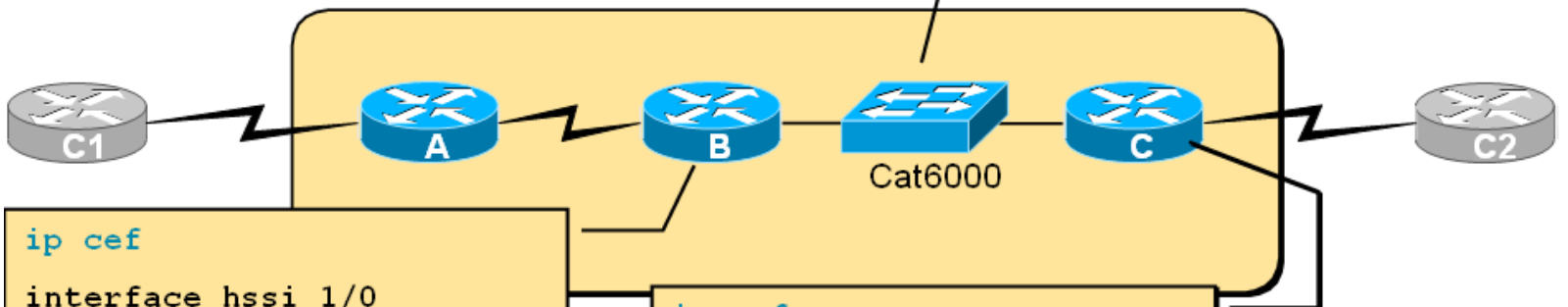
Use access lists to prevent customers from running TDP with your routers.

```
ip cef
interface serial 3/1
 ip access-group NoTDP in
interface hssi 1/0
 mpls ip
ip access-list NoTdp deny tcp any any eq 711
ip access-list NoTdp permit ip any any
```

MPLS on LAN Configuration Example

Jumbo frames have to be enabled on the switch.

```
set port 1/3 jumbo enable
set port 1/4 jumbo enable
```




```
ip cef
interface hssi 1/0
 mpls ip
interface fastethernet 0/0
 mpls ip
 mpls mtu 1512
```

```
ip cef
interface fastethernet 0/0
 mpls ip
 mpls mtu 1512
```

MPLS MTU is increased to 1512 to support 1500-byte IP packets and MPLS stack up to three levels deep.

Configuring IP TTL Propogation

router(config)#

no mpls ip propagate-ttl	**12.1(3)T**

- **By default, IP TTL is copied into label header at label imposition and label TTL is copied into IP TTL at label removal.**
- **This command disables IP TTL and label TTL propagation.**
  - **TTL value of 255 is inserted in the label header.**
- **The TTL propagation has to be disabled on ingress and egress edge LSR.**

sh ip cef detail

```
Router#show ip cef 192.168.20.0 detail
192.168.20.0/24, version 23, cached adjacency to Serial1/0.2
0 packets, 0 bytes
  tag information set
    local tag: 33
    fast tag rewrite with Se1/0.2, point2point, tags imposed: {32}
  via 192.168.3.10, Serial1/0.2, 0 dependencies
    next hop 192.168.3.10, Serial1/0.2
    valid cached adjacency
    tag rewrite with Se1/0.2, point2point, tags imposed: {32}
```

sh mpls ldp neighbor

```
Router#show tag-switching tdp neighbors
Peer TDP Ident: 192.168.3.100:0; Local TDP Ident
192.168.3.102:0
        TCP connection: 192.168.3.100.711 - 192.168.3.102.11000
        State: Oper; PIEs sent/rcvd: 55/53; ; Downstream
        Up time: 00:43:26
        TDP discovery sources:
          Serial1/0.2
        Addresses bound to peer TDP Ident:
          192.168.3.10    192.168.3.14    192.168.3.100
```

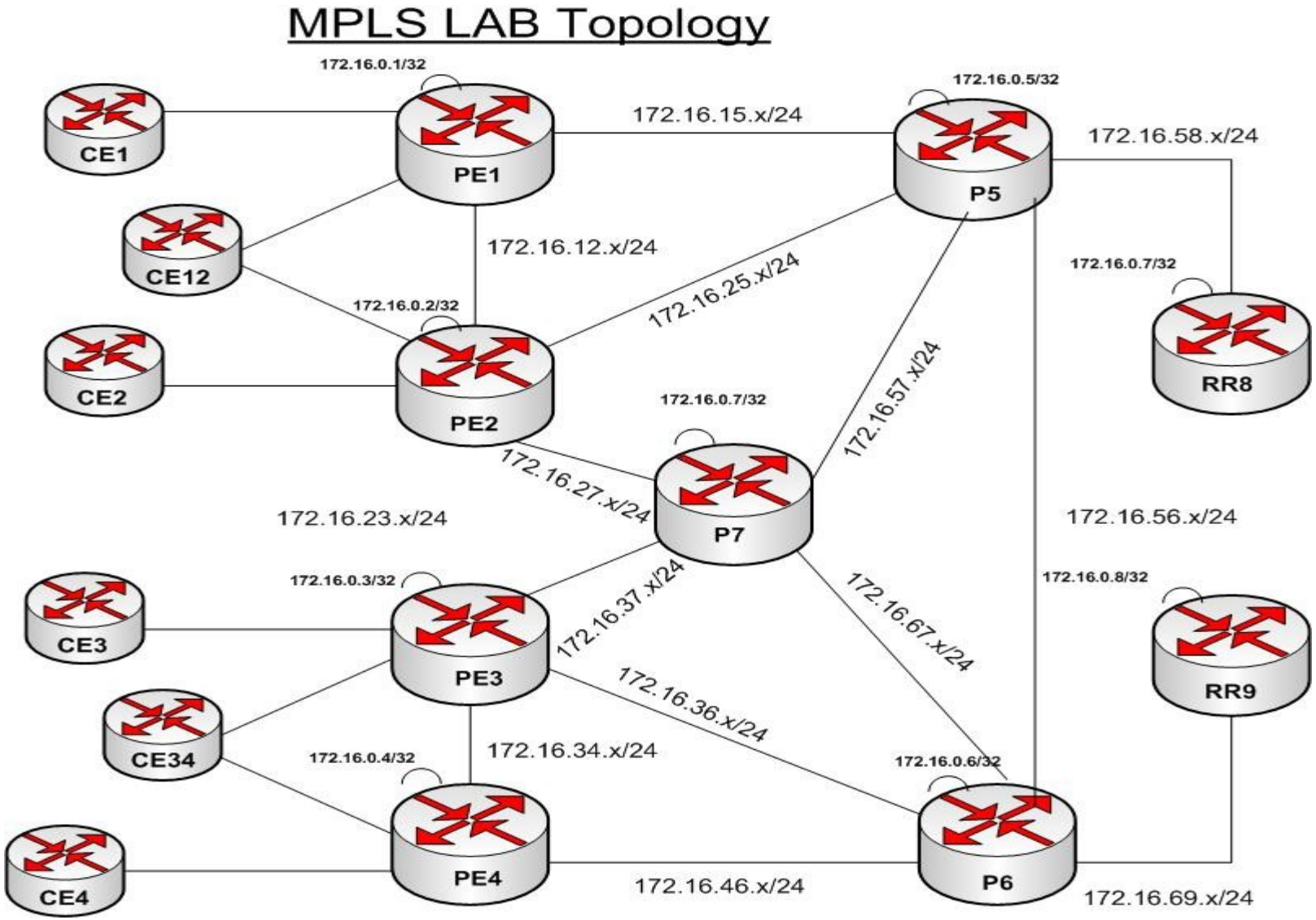
sh mpls ldp discovery

```
Router#show tag-switching tdp discovery
Local TDP Identifier:
    192.168.3.102:0
TDP Discovery Sources:
    Interfaces:
        Serial1/0.1: xmit/recv
            TDP Id: 192.168.3.101:0
        Serial1/0.2: xmit/recv
            TDP Id: 192.168.3.100:0
```

sh mpls forwarding table

```
Router#show tag-switching forwarding-table detail
Local  Outgoing    Prefix            Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     Untagged    192.168.3.3/32    0          Se1/0.3     point2point
       MAC/Encaps=0/0, MTU=1504, Tag Stack{}
27     Pop tag     192.168.3.4/32    0          Se0/0.4     point2point
       MAC/Encaps=4/4, MTU=1504, Tag Stack{}
       20618847
28     29          192.168.3.4/32    0          Se1/0.3     point2point
       MAC/Encaps=4/8, MTU=1500, Tag Stack{29}
       18718847 0001D000
```

MPLS LAB Topology

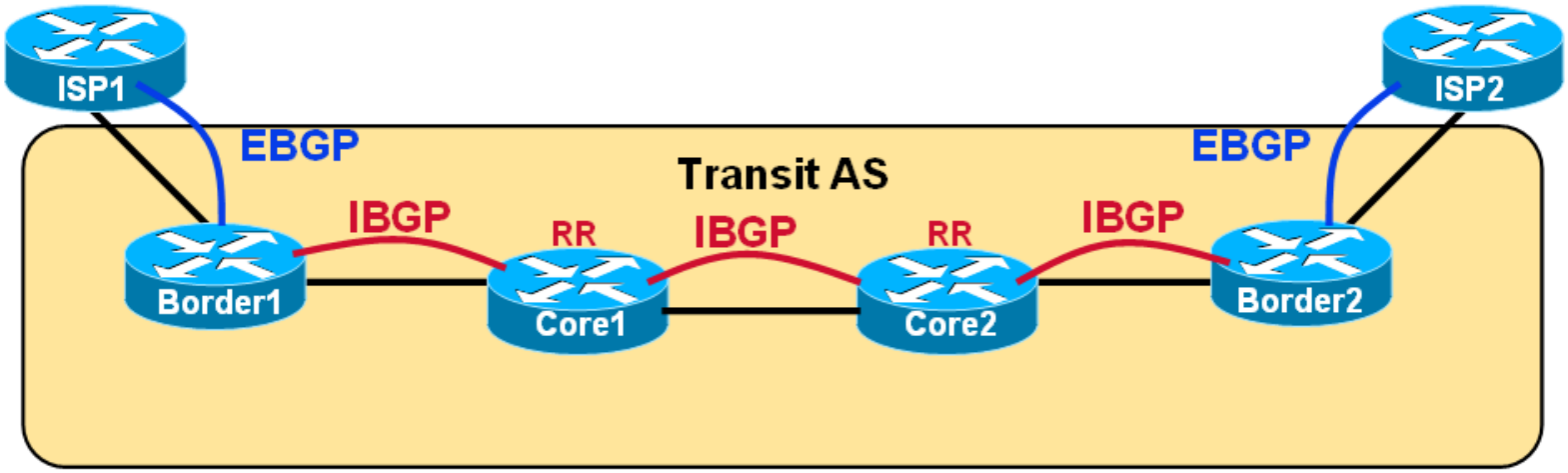


MPLS in BGP Transit AS

MPLS – BGP Interation

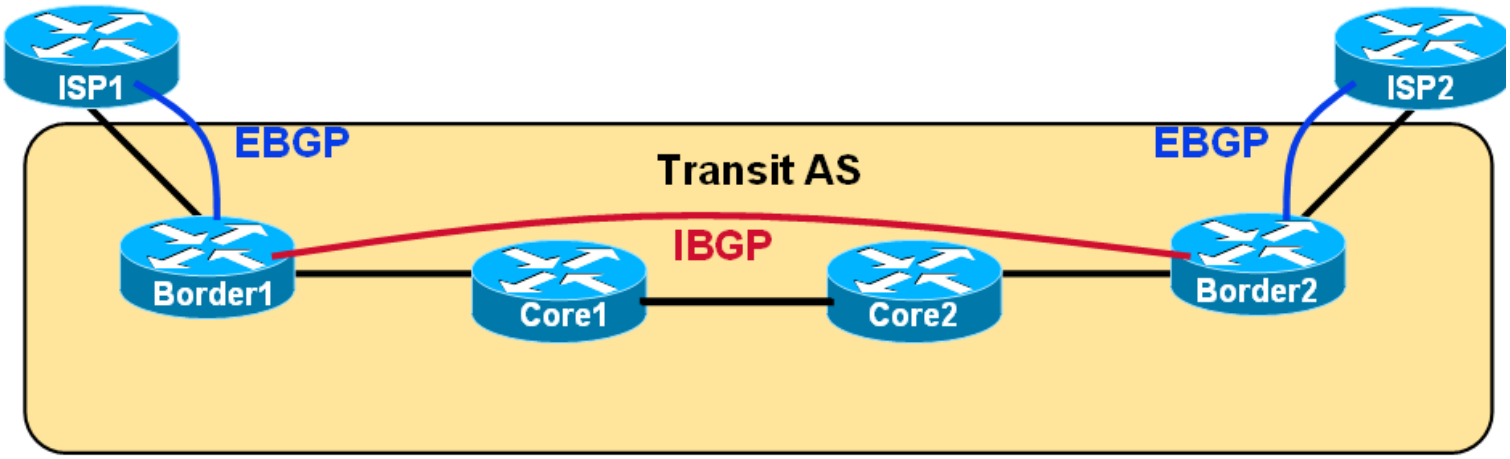
- Labels are assigned to FECs.
- FEC in unicast IP routing is equal to a destination prefix found in an IP routing table.
- This is true only for IGP-derived prefixes.
- BGP-derived prefixes are assigned the label that is used for the BGP next-hop address.
- Result: all prefixes learned from an external BGP neighbor use a single label.

Traditional BGP AS Design Requirement



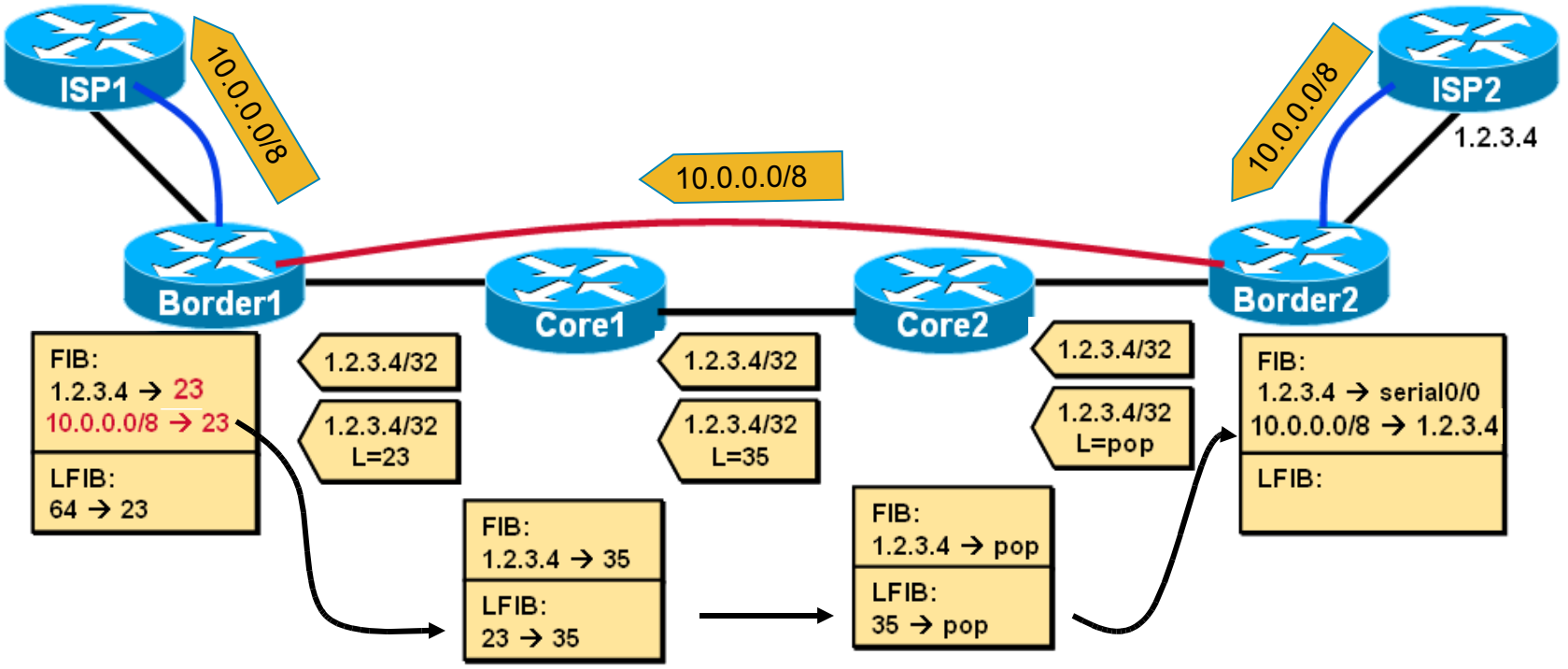
- All core routers are required to run BGP.
- All core routers require full Internet routing information (more than 250 000 networks) to be able to forward IP packets between ISP1 and ISP2.

Simplified BGP Design with MPLS



- Only border routers are required to run BGP.
- Core routers run an IGP to learn about BGP next-hop addresses.
- Core routers run LDP or TDP to learn about labels for next-hop addresses.

MPLS based Transit AS Building FIB and LFIB



All routers are capable of forwarding packets to external destinations

Border (edge) routers label and forward IP packets.

Core routers forward labeled packets.

Benefit of MPLS-Based Transit AS

- Simplified BGP topology (only AS edge routers are required to run BGP with full Internet routing).
- Core routers do not require a lot of memory 250 000 networks may require more than 50 MB of memory for the BGP table, IP routing table, and CEF's FIB table and distributed FIB tables).
- Changes in the Internet do not impact core routers.
- Allows private addresses (RFC 1918) to be used in the core if TTL propagation is disabled (traceroute across the AS will not show any private addresses).

Troubleshooting LDP

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
    - Load sharing
    - MTU issues
  - Troubleshooting Tips
  - Case Studies

Control Plane – Troubleshooting Tips

- Check for same label protocol to be configured on both sides of the interface

  "Sh mpls ldp discovery | inc ldp|tdp"

- Check whether correct local LSR_ID is used on both LSRs (sh mpls ldp disc)

  "sh mpls ldp discovery" – 2nd line in output

- Don't assume that the neighbor discovery means everything is good.

Control Plane – Troubleshooting Tips

- Check IP reachability to remote LSR_ID on both LSRs

  “ping <lsr_id>”

- Check for ACL or ICMP unreachable blockages
- Untagged outgoing label for /32 routes i.e. PEs’ loopbacks is almost always alarming.
- Check the label binding for a prefix on both LSRs

  “sh mpls ldp bind <prefix> <mask>”

Control Plane – Troubleshooting Tips

- Make sure the LDP filtering (if configured) is correctly setup via ACL

  “sh mpls ip bind advertisement-acl | inc Prefix”

- Good practice is to configure the Loopback0 as the router-ID for LDP

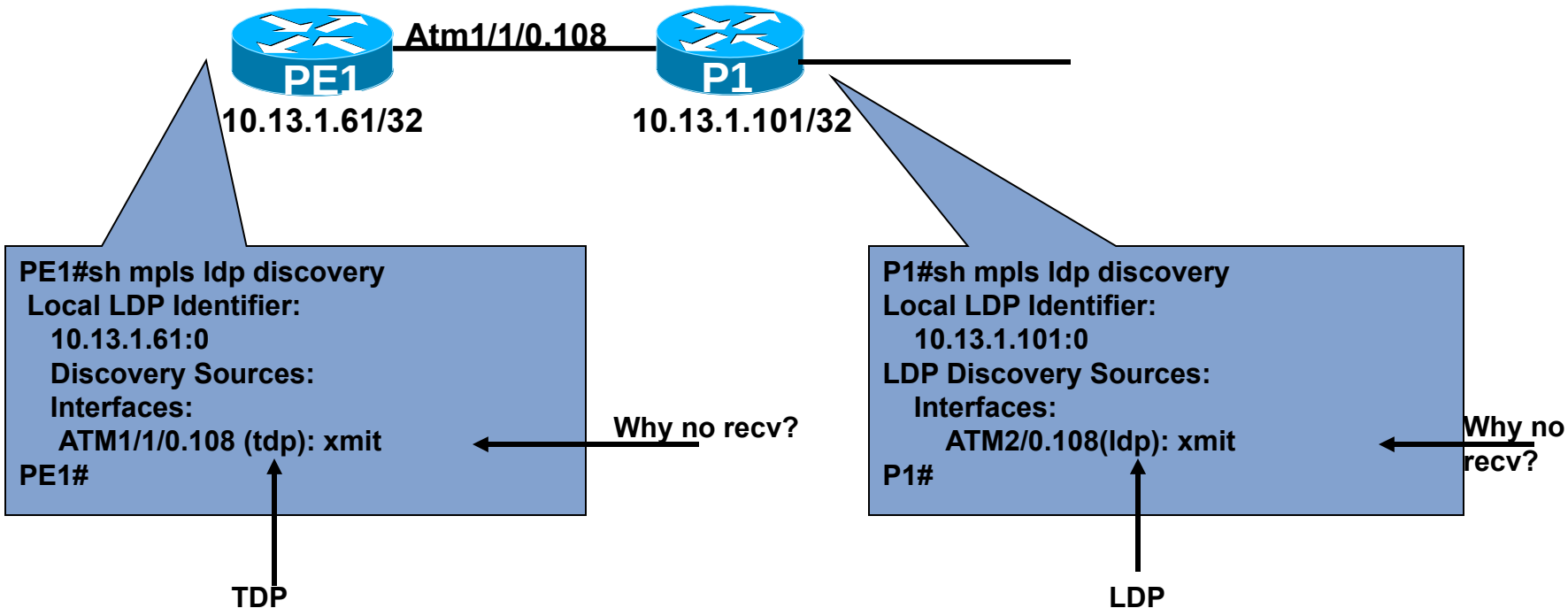
  “mpls ldp router-id loopback0 force”

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
    - Load sharing
    - MTU issues
  - Troubleshooting Tips
  - Case Studies

MPLS Control Plane – Protocol mismatch

Prob#1 – session establishment (Protocol mismatch)

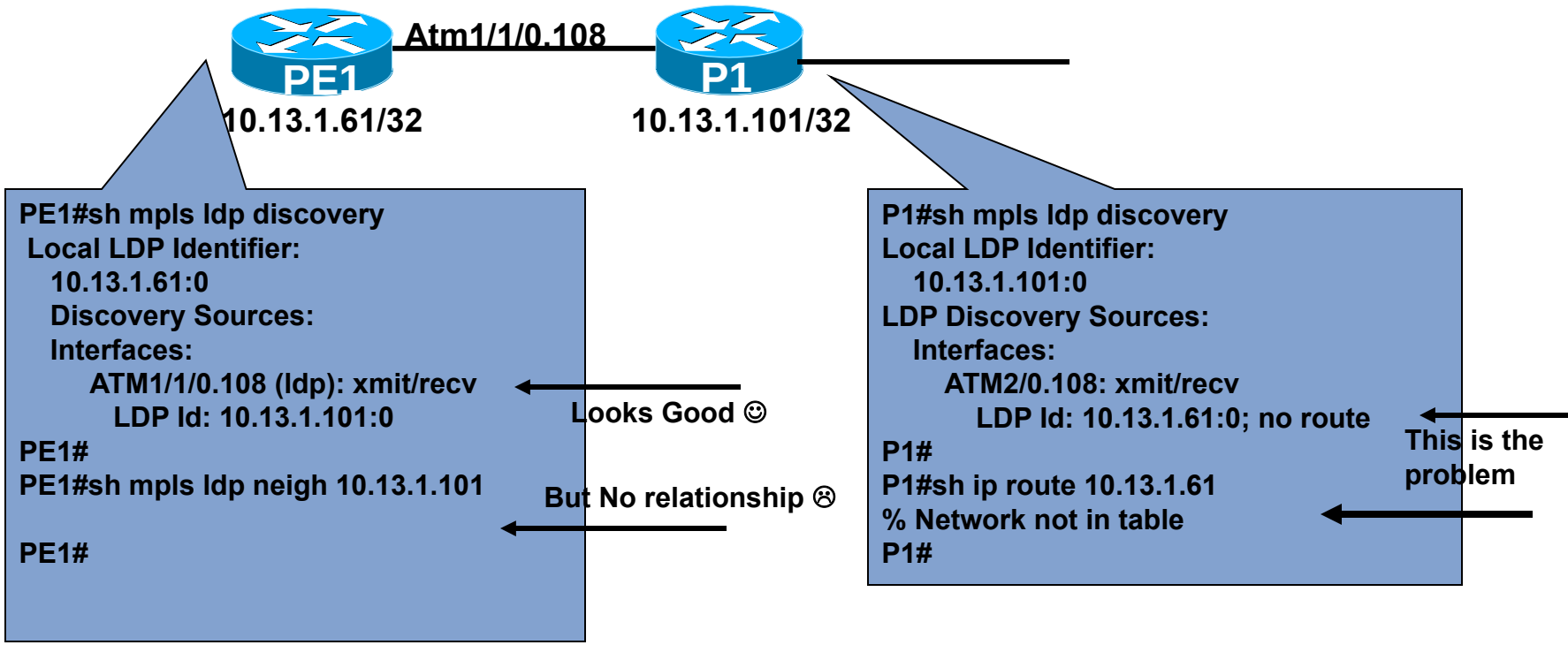


TIP – Check for the protocol mismatch and fix it.

```
PE1(config)#int atm1/1/0.108
PE1(config-if)#mpls label protocol ldp
```

MPLS Control Plane – No route

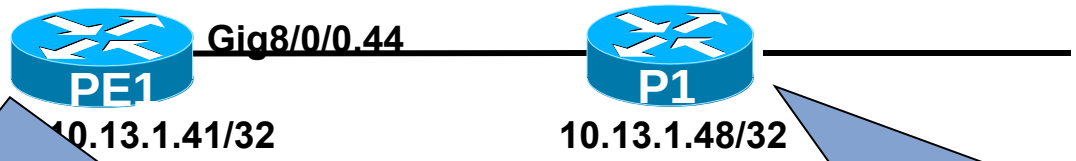
Prob#2 – session establishment (No route to peer)



TIP – Check for IP reachability to LDP_ID. Fix it by letting PE1 advertise 10.13.1.61/32 via IGP to P1.

MPLS Control Plane – No Specific route

Prob#3 - Session establishment (no specific route)



oops →

```
PE1#sh mpls ldp neighbor 10.13.1.48

PE1#
```

Ok. →

```
PE1#sh mpls ldp discovery
 Local LDP Identifier:
    10.13.1.41:0
        Gi8/0/0.44 (ldp): xmit/recv
           LDP Id: 10.13.1.48:0
PE1#
```

Ok. →

```
PE1#sh ip route 10.13.1.48
Routing entry for 10.13.1.48/32
  Known via "isis", distance 115, metric 10, type level-1
  Redistributing via isis
  Last update from 10.13.4.9 on Gig8/0/0.44, 20:22:14
ago
  Routing Descriptor Blocks:
  * 10.13.4.9, from 10.13.1.48, via Gigt8/0/0.44
      Route metric is 10, traffic share count is 1
PE1#
```

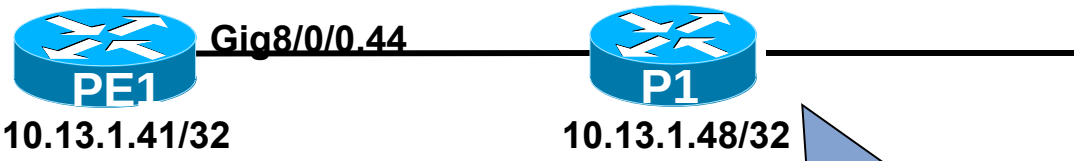
```
P1#sh mpls ldp neighbor 10.13.1.41

P1#
P1#sh mpls ldp discovery
 Local LDP Identifier:
    10.13.1.48:0
        Gi3/0/0.44 (ldp): xmit/recv
           LDP Id: 10.13.1.41:0
P1#
P1#sh ip route 10.13.1.41
Routing entry for 10.13.0.0/22
  Known via "bgp 30000", distance 200, metric 0
  Tag 1, type internal
  Last update from 10.13.1.251 20:10:38 ago
  Routing Descriptor Blocks:
  * 10.13.1.251, from 10.13.1.40, 20:10:38 ago
      Route metric is 0, traffic share count is 1
      AS Hops 5
P1#
```

Ouch

P1 doesn't have a specific route to PE1.

MPLS Control Plane – No Specific route (contd..)

Prob#3 - Session establishment (Contd)




```
PE1#ping 10.13.1.48
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.13.1.48, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
1/1/4 ms
PE1#
```

```
P1#ping 10.13.1.41
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.13.1.41,
timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
P1#
```

Eeeekks !! It is an IP problem.

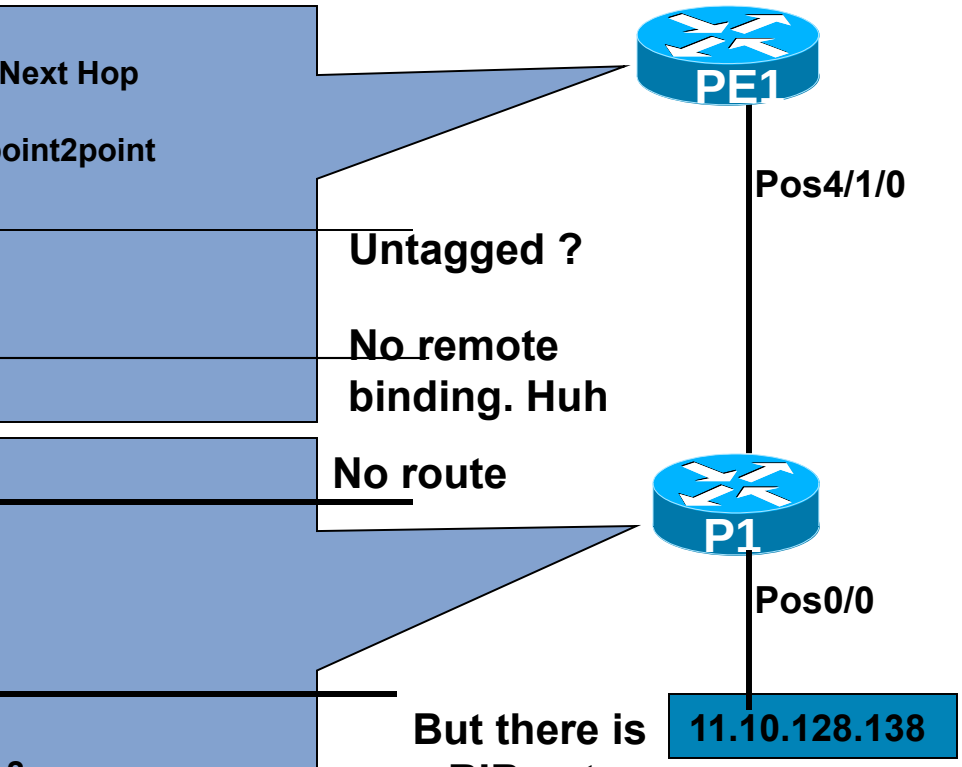
TIP – Check for IP connectivity first. Unless Layer3 is up, Layer4 (TCP session for LDP) won't come up.

MPLS Control Plane – Untagged outbound label

Prob#4 - "Untagged" problem

```
PE1#sh tag for 11.10.128.138
Local  Outgoing    Prefix          Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Untagged    11.10.128.138/32  0        PO4/1/0    point2point
PE1#

PE1#sh mpls ldp  bind 11.10.128.138 32
  tib entry: 11.10.128.138/32, rev 14
       local binding:  tag: 16
PE1#
```

```
P1#sh mpls ldp bind 11.10.128.138 32
  tib entry: 11.10.128.138/32, rev 4849(no route)
       local binding:  tag: 630
       remote binding: tsr: 10.13.1.54:0,   tag: 16
       remote binding: tsr: 11.10.65.12:0, tag: 48

P1#sh ip route 11.10.128.138
Routing entry for 11.10.0.0/16
  Known via "isis", distance 115, metric 44, type level-2
  Redistributing via isis
  Last update from 11.10.65.13 on POS0/0, 1d00h ago
  Routing Descriptor Blocks:
  * 11.10.65.13, from 11.10.128.31, via POS0/0
      Route metric is 44, traffic share count is 1
P1#
```



MPLS Control Plane – Untagged outbound label (contd..)

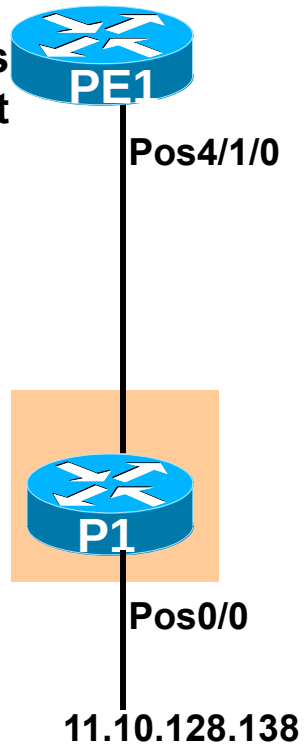
Prob#4 - "Untagged" problem (contd)

```
P1#sh ip cef 11.10.128.138
11.10.0.0/16, version 142, cached adjacency to POS0/0
0 packets, 0 bytes
  tag information set
    local tag: 307
    fast tag rewrite with PO0/0, point2point, tags imposed {48}
  via 11.10.65.13, POS0/0, 0 dependencies
    next hop 11.10.65.13, POS0/0
    unresolved        <<<<<<<<<<<<<<<<<<<<<<<<<<
    valid cached adjacency
    tag rewrite with PO0/0, point2point, tags imposed {48}
P1#
```

FIB's local label is different from that of LIB

Unresolved ?

```
P1#clear ip route 11.10.128.138
P1#sh mpls ldp bind 11.10.128.138 32
  tib entry: 11.10.128.138/32, rev 4849
        local binding:  tag: 307
        remote binding: tsr: 10.13.1.54:0, tag: 16
        remote binding: tsr: 11.10.65.20:0,tag:48
P1#
```



TIP – If local label for a prefix is not same in FIB and LIB, then issue "clear ip route <prefix>" to fix.

MPLS Control Plane – No LFIB entry

Prob#5 – LFIB entry disappears

- No LFIB entry
- This might occur if the RIB owner for an IPv4 routes changes from IGP to BGP
- LDP doesn't allocate labels for the BGP owned IPv4 routes
- Notice the absence of local binding in LIB for that route

MPLS Control Plane – No LFIB entry (contd..)

```
7206-PE-SOUTH-1#sh mpls ldp bind 4.4.0.0 24
  tib entry: 4.4.0.0/24, rev 152
        remote binding: tsr: 10.13.1.69:0, tag: 213
        remote binding: tsr: 10.13.1.68:0, tag: 212
7206-PE-SOUTH-1#
```

No Local Binding

```
7206-PE-SOUTH-1#sh ip route 4.4.0.0
Routing entry for 4.4.0.0/24
  Known via "bgp 30000", distance 200, metric 0
  Tag 1, type internal
  Redistributing via isis, ospf 1
  Last update from 10.13.1.251 5d17h ago
  Routing Descriptor Blocks:
  * 10.13.1.251, from 10.13.1.40, 5d17h ago
      Route metric is 0, traffic share count is 1
      AS Hops 5
      Route tag 1

7206-PE-SOUTH-1#
```

Because it is a BGP learned prefix

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
    - Load sharing
    - MTU issues
  - Troubleshooting Tips
  - Case Studies

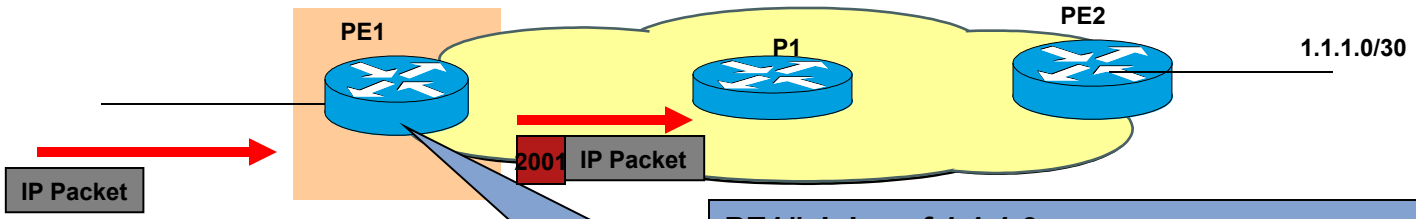
MPLS Forwarding Plane

- Three cases in the MPLS forwarding -
  Label Imposition - IP to MPLS conversion
  Label swapping - MPLS to MPLS
  Label disposition - MPLS to IP conversion

- So, depending upon the case, we need to check-
  FIB - For IP packets that get forwarded as MPLS
  LFIB - For MPLS packets that get fwded as MPLS
  LFIB - For MPLS packets that get fwded as IP

MPLS Forwarding Plane

Case 1: IP packets get forwarded as MPLS

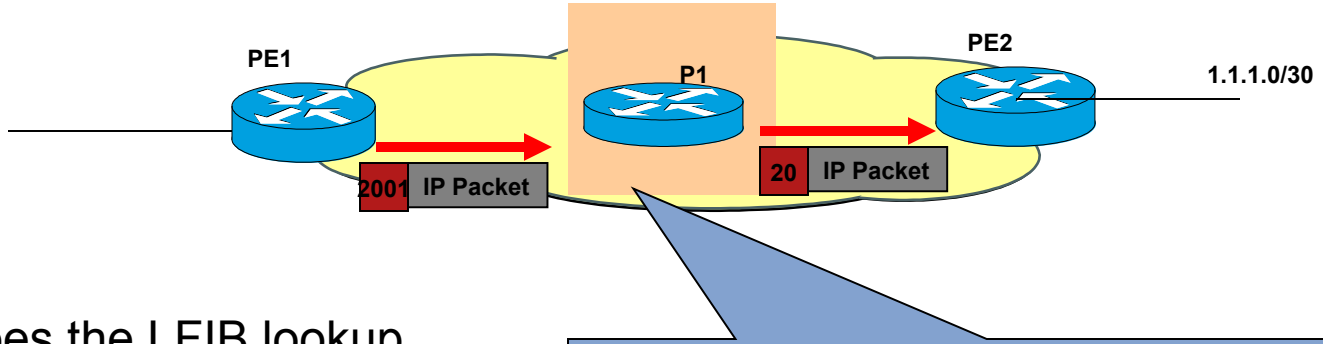


- PE1 does a FIB lookup for the incoming IP packet
- It imposes the label (if there is one)
- For troubleshooting, look at the FIB (not LFIB)

```
PE1#sh ip cef 1.1.1.0
1.1.1.0/30, version 25, epoch 0, cached adjacency 10.13.1.5
0 packets, 0 bytes
  tag information set
    local tag: 20
    fast tag rewrite with Et0/0, 10.13.1.5, tags imposed: {2001}
  via 10.13.1.5, Ethernet0/0, 0 dependencies
    next hop 10.13.1.5, Ethernet0/0
    valid cached adjacency
    tag rewrite with Et0/0, 10.13.1.5, tags imposed: {2001}
PE1#
```

MPLS Forwarding Plane

Case 2: MPLS packets get forwarded as MPLS



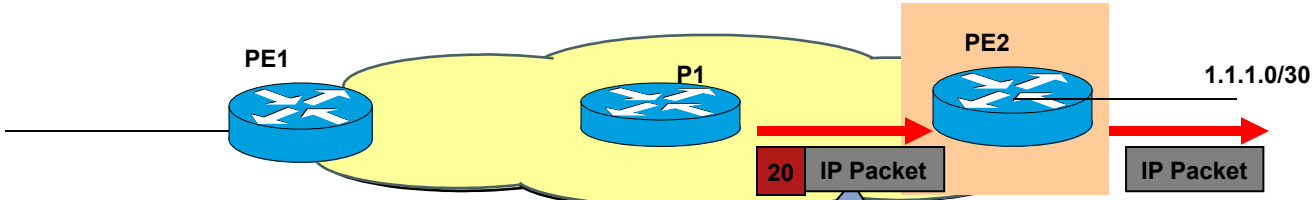
- P1 does the LFIB lookup for incoming MPLS packets
- P1 could swap (or dispose) the label
- For troubleshooting, look at the LFIB (not FIB)

```
P1#sh mpls for 1.1.1.0
Local  Outgoing    Prefix          Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id    switched   interface
2001   20     1.1.1.1.0/30    0        Se2/0      point2point
P1#
```

```
P1#sh mpls for 10.13.1.62
Local  Outgoing    Prefix          Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id    switched   interface
2001   Pop tag     10.13.1.62/32    0        Se2/0      point2point
P1#
```

MPLS Forwarding Plane

Case 3: MPLS packets get forwarded as IP



- Typically happen at the edge.
- Could also happen at the PHP router
- For troubleshooting, look at the LFIB (not FIB)

```
PE2#sh mpls for 1.1.1.0
Local  Outgoing    Prefix          Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id    switched   interface
20     Untagged    1.1.1.1.0/30    0          Se2/0       point2point
PE2#
```

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
  - Load sharing
  - MTU issues
  - Troubleshooting Tips
  - Case Studies

MPLS Forwarding Plane - Loadsharing

- Loadsharing (due to multiple paths to a prefix) in MPLS is no different from that of IP
- Hashing-algorithm is still the typical 'FIB based' i.e per-dest loadsharing by default **
- So the below "show command" is still relevant

  "Sh ip cef exact-route <source> <dest>" etc.

- But the dest must be known in the FIB table, otherwise the command won't work.

  Won't work on P routers for the VPN prefixes.

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
    - Load sharing
    - MTU issues
  - Troubleshooting Tips
  - Case Studies

MPLS Fwd Plane - Fragmentation

- After the Layer2 header is added to the IP packet, the resulting packet size shouldn't exceed the max packet size (IP MTU size) applicable . Otherwise, packet will be fragmented.
- MTU size needs to be tuned to avoid fragmentation in MPLS network
- MTU could be increased only for MPLS packets => MPLS MTU

Fragmentation
MTU Setting in MPLS

- Most of the interfaces (depending upon the hardware) support transmitting packets bigger than the “interface MTU” size
- “mpls mtu <bytes>” can be applied to an interface to change the MPLS MTU size on the interface
- MPLS MTU size is checked by the router

  while converting an IP packet into a labeled packet or transmitting a labeled packet

Fragmentation
MTU Setting in MPLS

- Remember that -
- ‘mpls mtu <bytes>” command has no effect on “interface or IP MTU” size.
- By default, MPLS MTU = interface MTU
- MPLS MTU setting doesn’t affect MTU handling for IP-to-IP packet switching

MTU Setting in MPLS
Configuring the MPLS MTU

```
RSP-PE-WEST-4(config)#int fa1/1/0
RSP-PE-WEST-4(config-if)#mpls mtu 1508
RSP-PE-WEST-4(config-if)#^Z
RSP-PE-WEST-4#
```

MTU Setting in MPLS
Before setting the MPLS MTU

- Interface MTU is 1500 bytes (no change):

```
RSP-PE-WEST-4#sh int fa1/1/0
  FastEthernet1/1/0 is up, line protocol is up
  Hardware is cyBus FastEthernet Interface, address is 0004.4e75.4828
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
  ……..……
RSP-PE-WEST-4#
```

- MPLS MTU is 1508 bytes (changed):

```
RSP-PE-WEST-4#sh mpls interface fa1/1/0 detail
Interface FastEthernet1/1/0:
        IP tagging enabled
        TSP Tunnel tagging not enabled
       Tagging operational
        ……..……..
        MTU = 1508
RSP-PE-WEST-4#
```

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
    - Load sharing
    - MTU issues
  - Troubleshooting Tips
  - Case Studies

MPLS Fwd Plane – Troubleshooting Tips

- If PXF based platform, then check the PXF[1]
- On distributed platforms, check the FIB/LFIB entries on the LC
- On distributed platforms that have HW-based forwarding, check the FIB/LFIB on specific HW i.e. PSA (E2), Alpha(E3) on GSR etc.

  Sh ip psa-cef, sh tag psa-tag, sh ip alpha-cef etc

[1] Not all PXF based platform support MPLS; they punt the MPLS packets to the CEF path.

MPLS Fwd Plane – Troubleshooting Tips

- Label imposition is always done using FIB
- Label swapping and disposition is always done using LFIB
- Increase the MPLS MTU to accommodate the largest packet payload size
- Make sure that baby giant/jumbo is enabled on the Ethernet switches

MPLS Fwd Plane – Troubleshooting Tips

- Check that MPLS enabled interface has “TAG” adjacency via

  “sh adjacency <interface>”

- Check that the LFIB’s outgoing label is same as the incoming label in neighbor’s LFIB
- Check the LSP via traceroute that shows labels used by each router in the path **

  “traceroute <prefix>”

MPLS Forwarding Plane – TAG adj

- Make sure that the interface has the “tag” adjacency along with “IP” adj, otherwise MPLS packets will not get switched on that interface

```
PE1#sh adjacency e0/0 de
Protocol  Interface                 Address
TAG       Ethernet0/0               10.13.1.5(6)
                                    0 packets, 0 bytes
                                    AABBCC006500AABBCC0001008847
                                    mpls adj   never
                                    Epoch: 0
IP        Ethernet0/0               10.13.1.5(35)
                                    0 packets, 0 bytes
                                    AABBCC006500AABBCC0001000800
                                    ARP        03:46:13
                                    Epoch: 0
PE1#
```

L2 header for MPLS

L2 header for IP

MPLS Fwd Plane – Show commands

- “sh mpls forwarding”

  Shows all LFIB entries (vpn, non-vpn, TE etc.)

- “sh mpls forwarding <prefix>”

  LFIB lookup based on a prefix

- “sh mpls forwarding label <label>”

  LFIB lookup based on an incoming label

- “sh mpls forwarding <prefix> detail”

  Shows detailed info such as L2 encap etc

MPLS Fwd Plane – Debugs

- Be Careful on the production routers

- "Debug mpls lfib cef"

  Useful for seeing FIB and LFIB interaction when a label is missing for a prefix

- "debug mpls lfib struct"

  Shows changes in the LFIB structures when label is allocated/deallocated

Agenda

- Control Plane
  - Troubleshooting Tips
  - Case Studies
- Forwarding Plane
  - Types of forwarding cases
    - Load sharing
    - MTU issues
  - Troubleshooting Tips
  - Case Studies

MPLS Forwarding Plane - No entry in LFIB

Prob#1 - No entries in LFIB

```
P1#sh mpls forwarding-table 10.13.1.61
Tag switching is not operational.
CEF or tag switching has not been enabled.
Local  Outgoing    Prefix            Bytes tag  Outgoing    Next Hop

tag    tag or VC   or Tunnel Id      switched   interface
```

```
P1#sh mpls ip binding
 10.13.1.61/32
        out label:    imp-null  lsr: 10.13.1.61:0
        out label:    21        lsr: 10.13.1.62:0
  10.13.1.62/32
        out label:    imp-null  lsr: 10.13.1.62:0
        out label:    17        lsr: 10.13.1.61:0
  10.13.1.101/32
        out label:    19        lsr: 10.13.1.62:0
        out label:    18        lsr: 10.13.1.61:0
  10.13.2.4/30
        out label:    imp-null  lsr: 10.13.1.62:0
        out label:    19        lsr: 10.13.1.61:0
P1#
```

```
P1#sh ip cef
%CEF not running
Prefix              Next Hop            Interface
P1#
```

TIP – Enable CEF. It is must for MPLS.

MPLS Forwarding Plane- Out label is Untagged

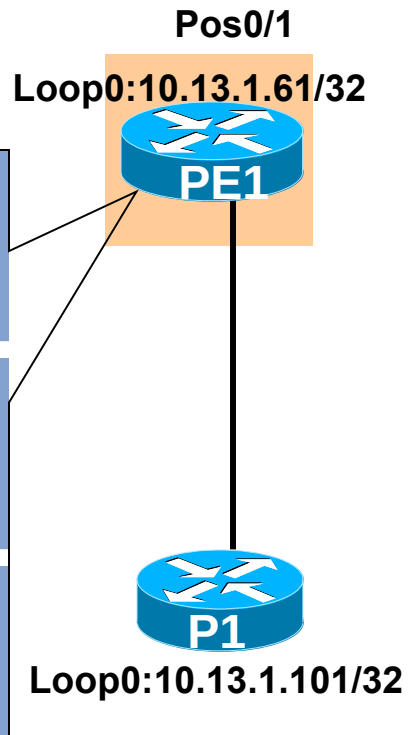
Prob#2 - "Untagged" problem

- LDP session is UP;LIB has correct binding;but LFIB has "Untagged" ☹

```
PE1#sh mpls for 10.13.1.101
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
20     Untagged    10.13.1.101/32   0          PO0/1       point2point
PE1#
```

```
PE1#sh mpls ip bind 10.13.1.101 32
  10.13.1.101/32
        in label:     20
        out label:    imp-null  lsr: 10.13.1.101:0
PE1#
```

```
PE1#sh adjacency pos0/1
Protocol Interface                 Address
TAG      POS0/1                    point2point(7) (incomplete)  <<=====Oops
IP       POS0/1                    point2point(39)
PE1#
```



TAG ADJ for pos0/1 is incomplete. No good.

MPLS Forwarding Plane- Out label is Untagged (contd..)

- **Adj is incomplete;check the interface.**

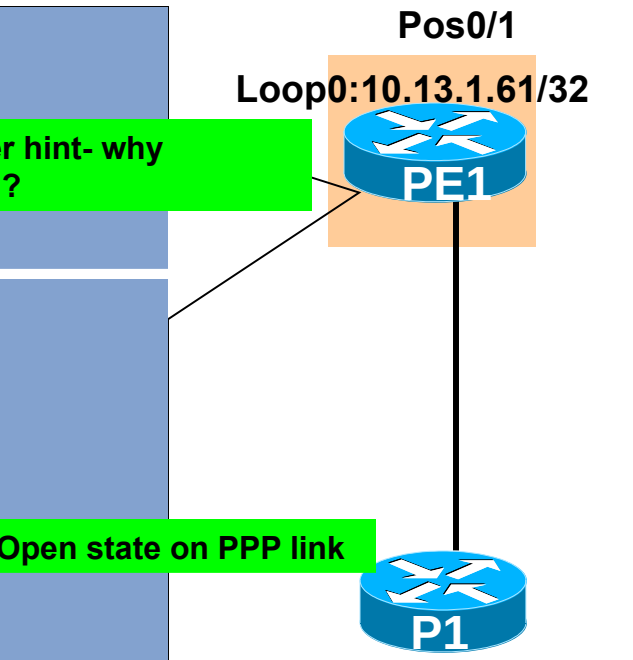
(contd)

```
PE1#sh mpls for 10.13.1.101 detail
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
12318  Untagged    10.13.1.101/32   0          PO0/1       point2point
          MAC/Encaps=0/0, MRU=4474, Tag Stack{}
       No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
PE1#
```

<<====== Another hint- why MAC/Encap is 0/0?

```
PE1#sh int pos0/1
POS0/1 is up, line protocol is up
  Hardware is Packet over SONET
  Description: OC48 to Redback
  Internet address is 10.1.17.1/24
  MTU 4470 bytes, BW 2488000 Kbit, DLY 100 usec, rely 255/255, load 1/255
  Encapsulation PPP, crc 32, loopback not set
  Keepalive not set
  Scramble disabled
  LCP Open
  Listen: TAGCP, CDPCP
  Open: IPCP
  Last input 00:00:01, output 00:00:03, output hang never
....
PE1#
```

<<========== TAGCP should also be in the Open state on PPP link



MPLS Forwarding Plane- Out label is Untagged (contd..)

(contd)

```
PE1#deb mpls adj
PE1#deb mpls lfib enc
PE1#
01:43:19: LFIB: finish res:inc tag=12318,outg=Imp_null,next_hop=0.0.0.0,POS0/1
01:43:19: LFIB: get ip adj: addr=0.0.0.0,is_p2p=1,fibidb=POS0/1,linktype=7
01:43:19: LFIB: get tag adj: addr=0.0.0.0,is_p2p=1,fibidb=POS0/1,linktype=90 INCOMPLETE
01:43:19: TAG ADJ: check 0.0.0.0, POS0/1 (537CF240/537CEE80)
01:43:19: LFIB: get ip adj: addr=0.0.0.0,is_p2p=1,fibidb=POS0/1,linktype=7
01:43:19: LFIB: get tag adj: addr=0.0.0.0,is_p2p=1,fibidb=POS0/1,linktype=90
01:43:19: LFIB: encaps:zero encaps,enc=0,mac=0,tag_adj POS0/1,itag=12318
```

TIP – If the interface doesn't have "TAG" adj, then the label will not get installed in LFIB. Fix PPP in this case.

MPLS Forwarding Plane- Recursive rewrite

Prob#3 - "Recursive rewrite" problem

- If you ever see "Recursive rewrite via…" in the "sh ip cef .." output, then it might indicate a problem.

```
2611-CE-30#sh ip cef 10.13.1.74
10.13.1.74/32, version 43, epoch 0, cached adjacency 5.5.5.14
0 packets, 0 bytes
  tag information set
    local tag: BGP route head
    fast tag rewrite with
    Recursive rewrite via 217.60.217.2/32, tags imposed {23}
  via 217.60.217.2, 0 dependencies, recursive
    next hop 5.5.5.14, Ethernet0/0.2 via 217.60.217.2/32
    valid cached adjacency
    tag rewrite with
    Recursive rewrite via 217.60.217.2/32, tags imposed {23}
2611-CE-30#
```

Problem with the 217.60.217.2. Check its label binding in FIB/LIB.

MPLS Forwarding Plane- Recursive rewrite (contd..)

(contd)

- "Recursive rewrite" usually means that
  - (a) Either the label to the next-hop is not available
  - (b) Or there is an internal problem with the CEF recursion resolution process
- (a) usually turns out to be a LDP problem, and should be fixed by investigating into LDP
- (b) could be fixed by "clear ip route <prefix>" or "clear ip bgp *"

MPLS Forwarding Plane- Recursive rewrite (contd..)

(contd)

- In order to troubleshoot (a), check the label availability for the next-hop (in LIB). If it is missing, then fix LDP.

```
2611-CE-30#sh mpls for 217.60.217.2
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
17     Untagged    217.60.217.2/32  0          Et0/0.2     5.5.5.14
2611-CE-30#
```

Untagged outgoing label

```
2611-CE-30#sh mpls ldp bind 217.60.217.2 32
  tib entry: 217.60.217.2/32, rev 14
        local binding:  tag: 17
2611-CE-30#
```

No remote label binding in LIB

```
2611-CE-30#sh mpls ldp dis
 Local LDP Identifier:
    217.60.217.3:0
    Discovery Sources:
    Interfaces:
        Ethernet0/0.2 (ldp): xmit
2611-CE-30#co
```

Because there is no LDP neighbor.

MPLS Forwarding Plane- Recursive rewrite (contd..)

(contd)

- LDP session needs to be established first.
- It is an LDP (control plane) problem.
- Troubleshoot for the LDP (as shown in the control plane section)

Conclusion

- Break down troubleshooting into systematic steps
- Look at things from a control plane and a forwarding plane perspective
- Do not panic

Introduction to VPN

Outline

Overview

Traditional Router-Based Networks

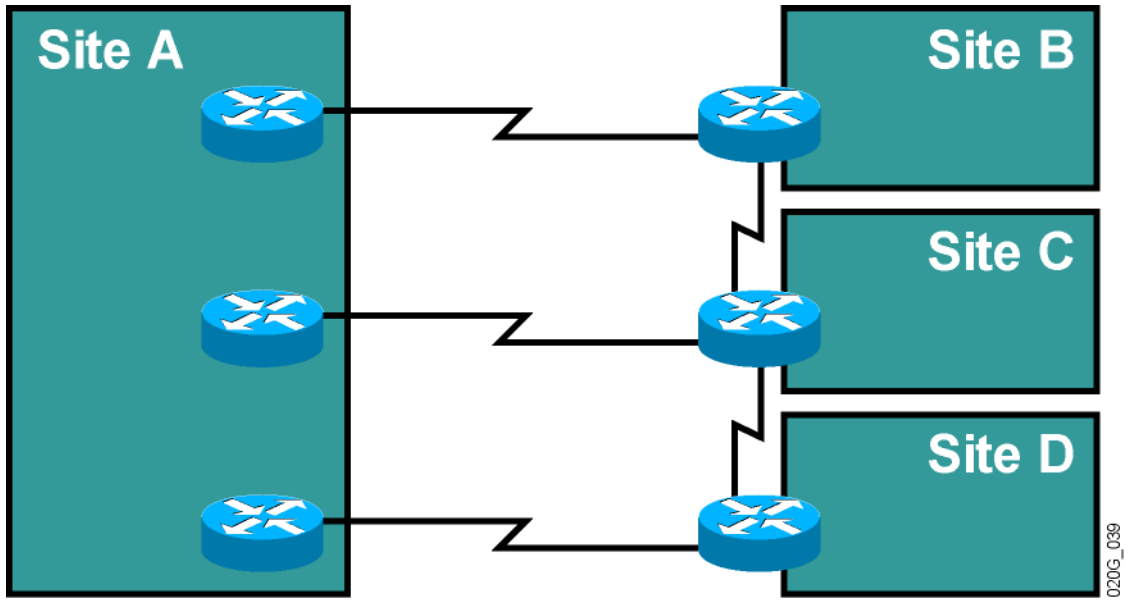
Virtual Private Networks

VPN Terminology

Switched WANs VPN Terminology

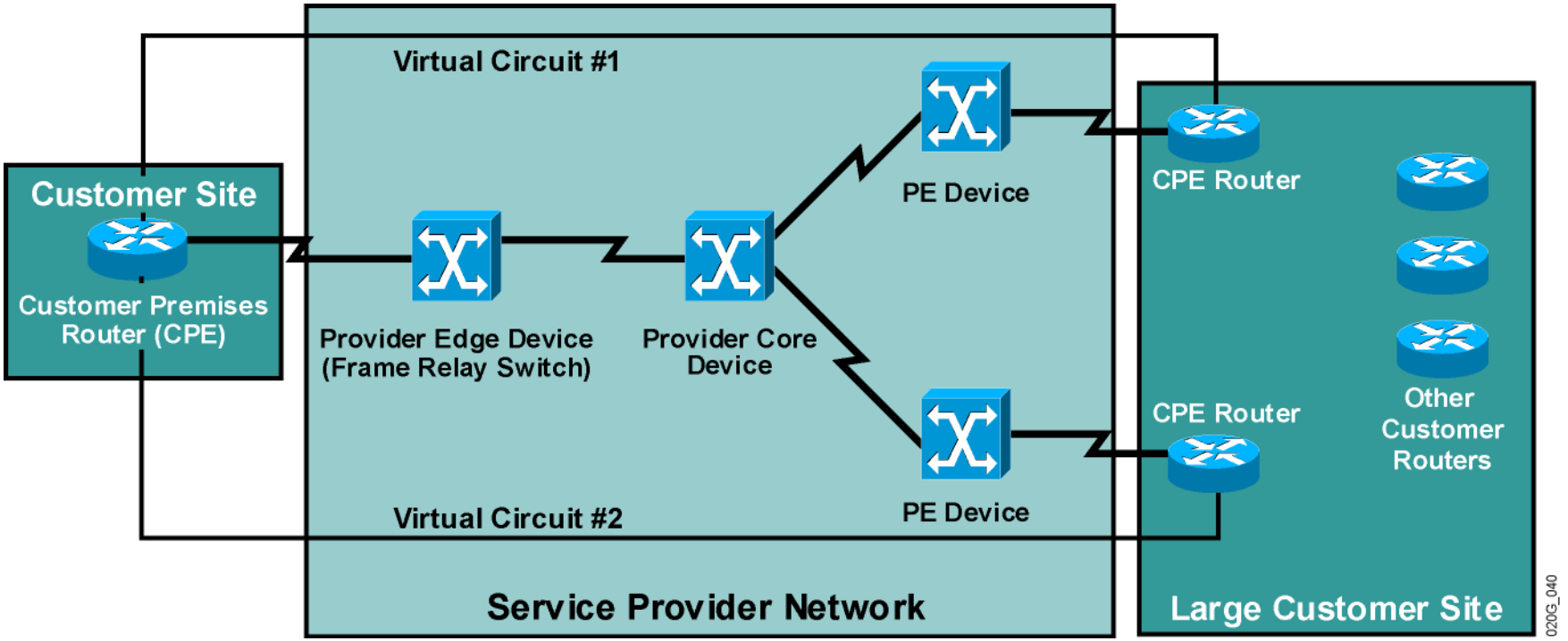
Lesson Summary

Traditional Router-Based Networks



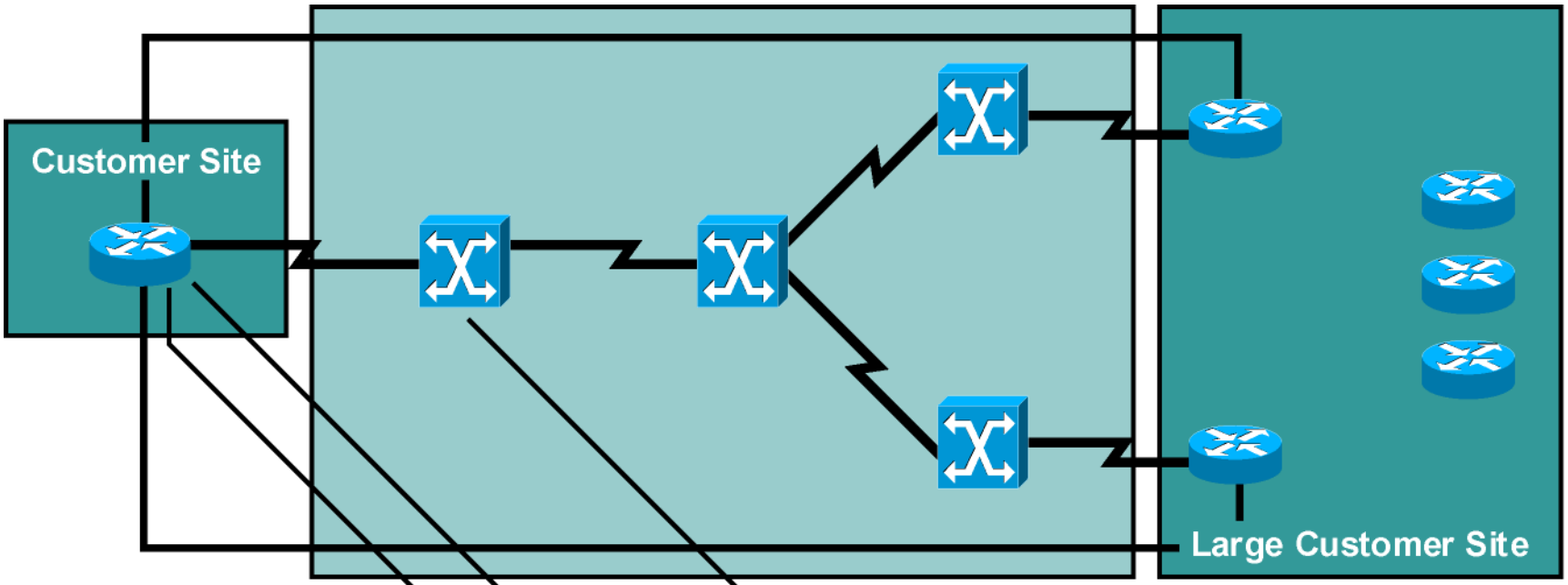
Traditional router-based networks connect customer sites through routers connected via dedicated point-to-point links.

Virtual Private Networks



- VPNs replace dedicated point-to-point links with emulated point-to-point links sharing common infrastructure.
- Customers use VPNs primarily to reduce their operational costs.

VPN Terminology



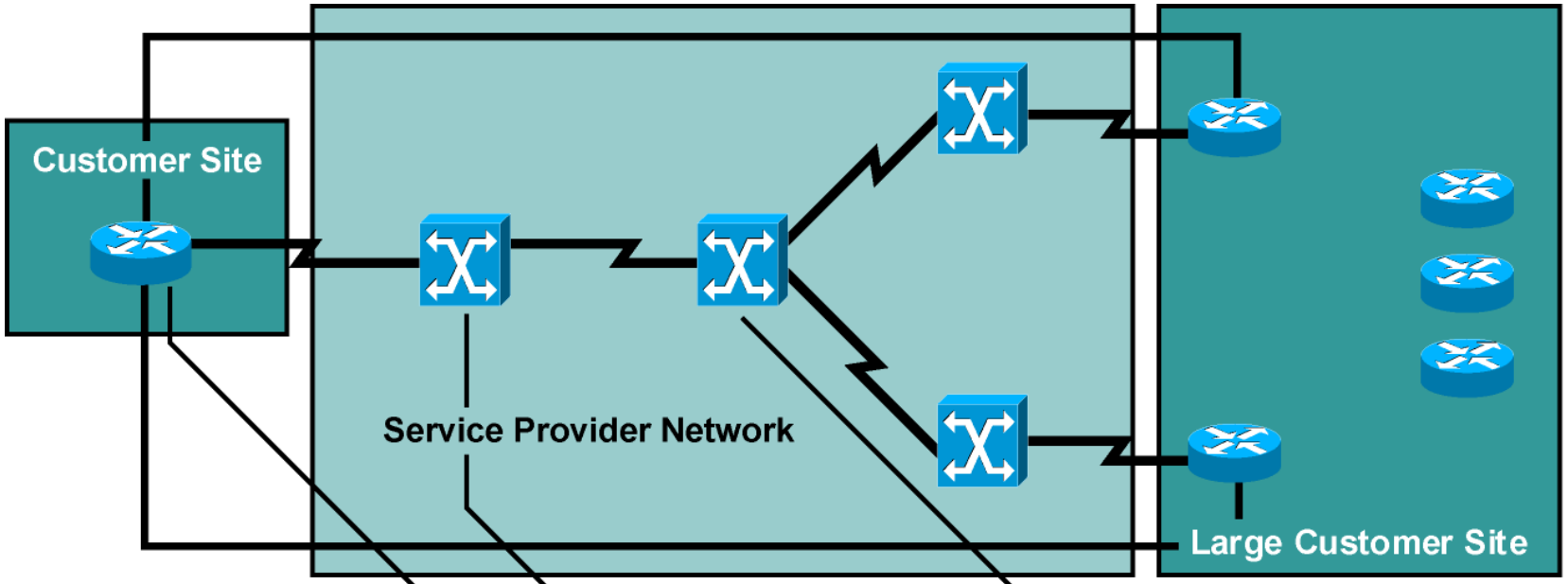
Provider network (P-network): The service provider infrastructure used to provide VPN services

Customer network (C-network): The part of the network still under customer control

Customer site: A contiguous part of the customer network (can encompass many physical locations)

020G_041

VPN Terminology (Cont.)



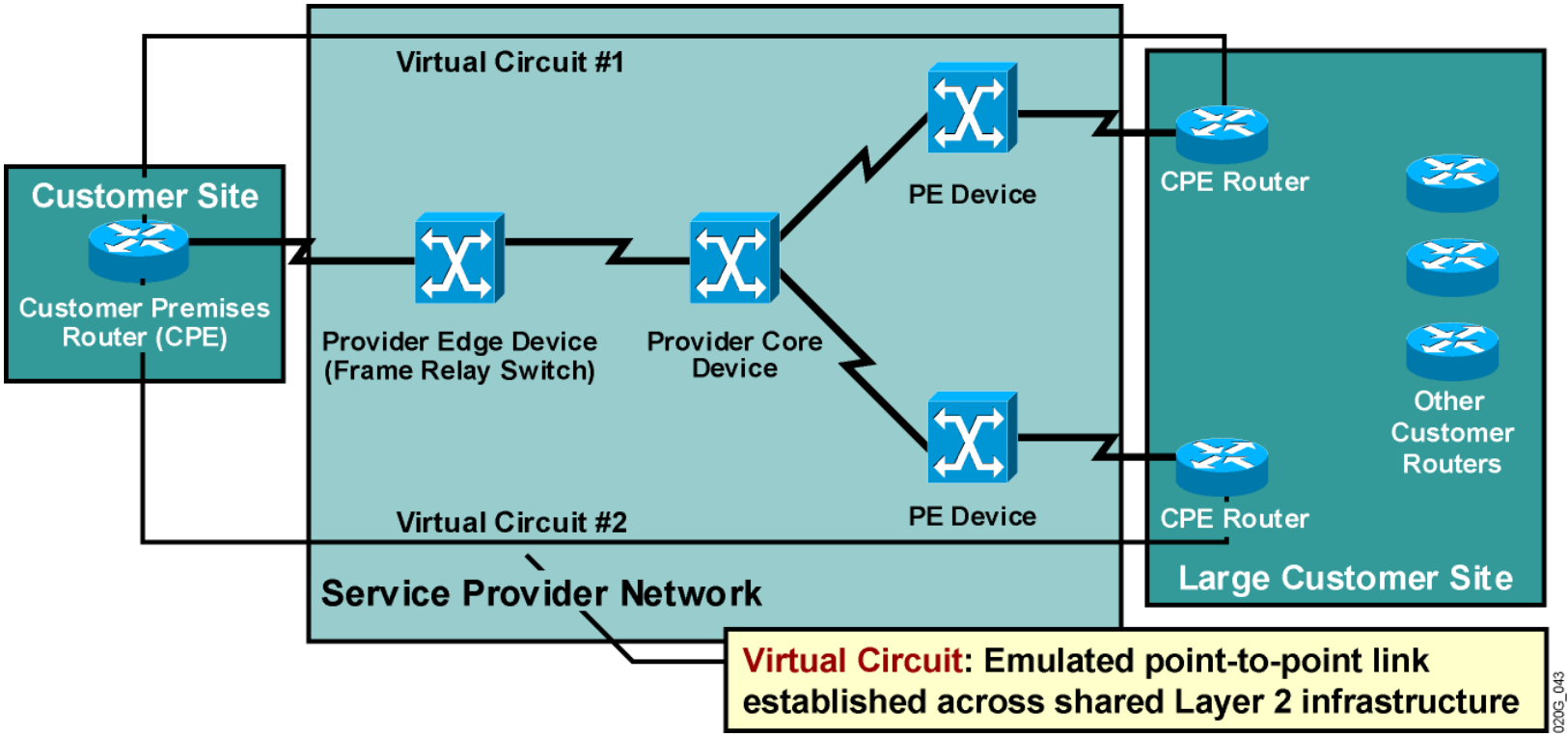
Provider (P) device: The device in the P-network with no customer connectivity

Provider edge (PE) device: The device in the P-network to which the CE devices are connected

Customer edge (CE) device: The device in the C-network that links to the P-network; also called customer premises equipment (CPE)

020G_042

Switched WANs VPN Terminology



- A PVC is established through out-of-band means (network management) and is always active.
- An SVC is established through CE-PE signaling on demand from the CE device.

Summary

Traditional router-based networks connect customer sites through routers connected via dedicated point-to-point links.

VPNs replaced dedicated point-to-point links with emulated point-to-point links sharing a common infrastructure.

Device names based on their position in the network are as follows:

- CE
- PE
- P

A PVC is established and is always active. An SVC is established through CE-PE signaling on demand from the CE device.

MPLS Bootcamp

Overlay and Peer-to-Peer VPNs

Outline

Overview

VPN Implementation Technologies

Overlay VPNs

Peer-to-peer VPNs

Benefits of VPN Implementations

Drawbacks of Various VPN Implementations

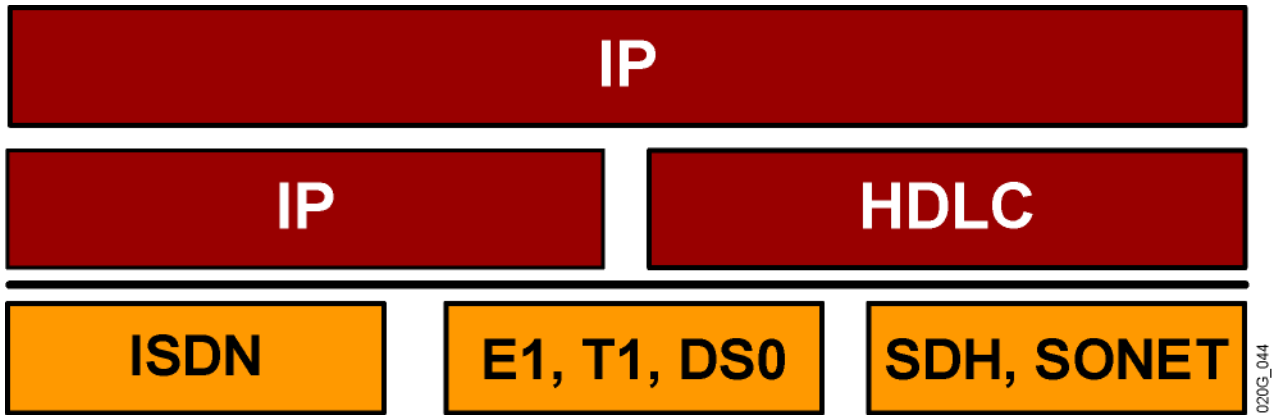
Drawbacks of Traditional Peer-to-Peer VPNs

Lesson Summary

VPN Implementation Technologies

- VPN services can be offered based on two major models:
  - Overlay VPNs, in which the service provider provides virtual point-to-point links between customer sites
  - Peer-to-peer VPNs, in which the service provider participates in the customer routing

Overlay VPNs
Layer 1 Implementation

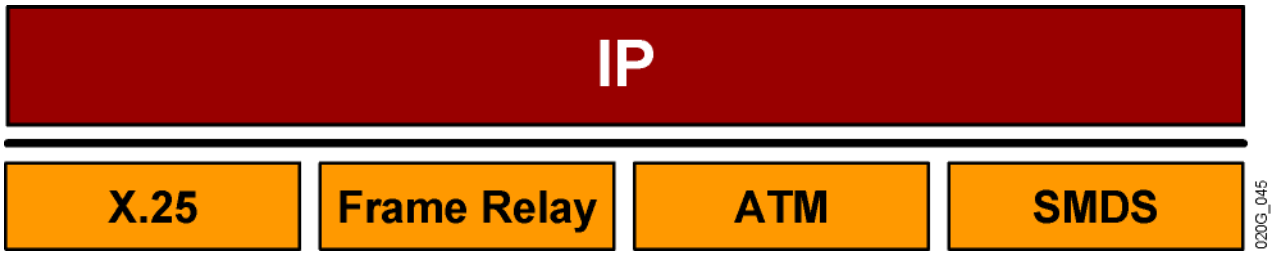


- This is the traditional TDM solution:

Service provider establishes physical-layer connectivity between customer sites.

Customer is responsible for all higher layers.

Overlay VPNs (Cont.)
Layer 2 Implementation

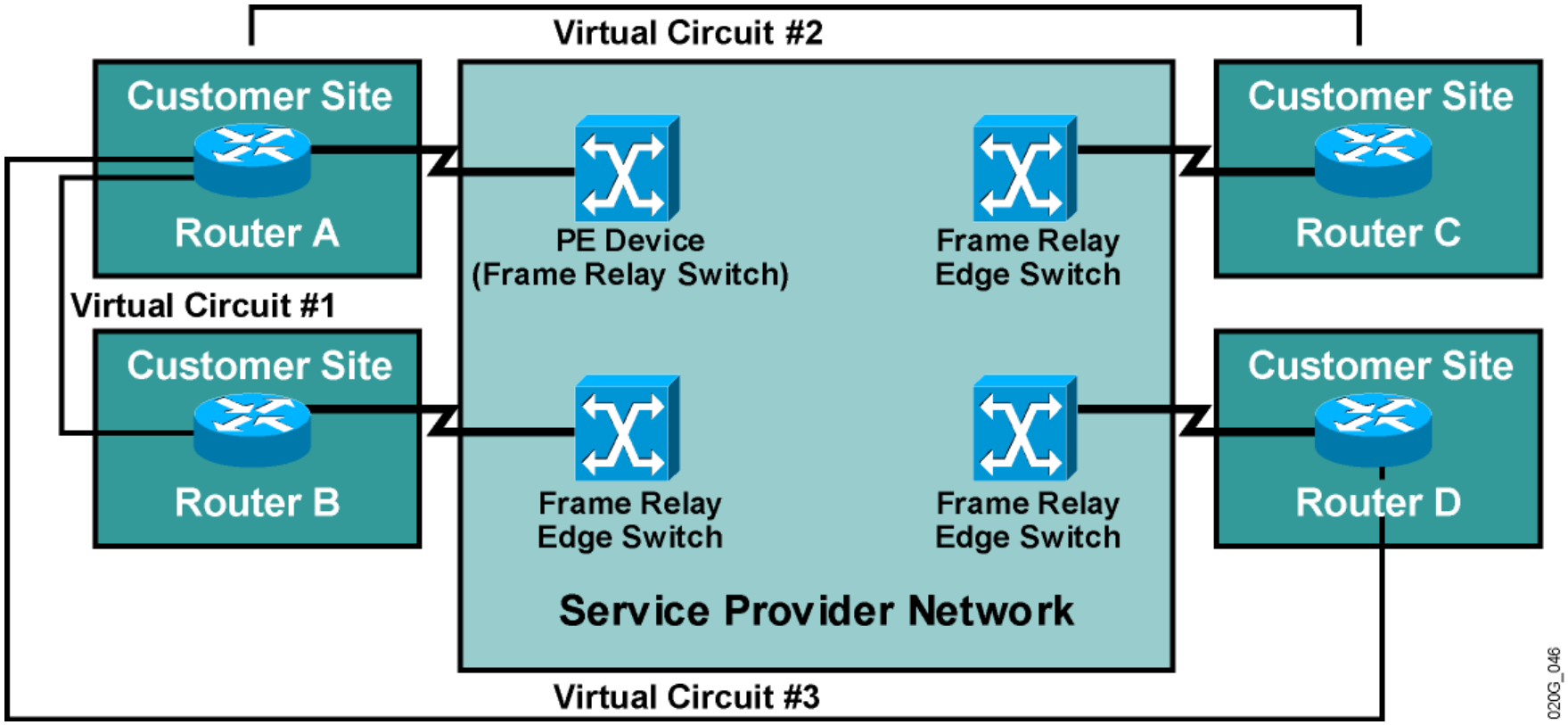


- This is the traditional switched WAN solution:

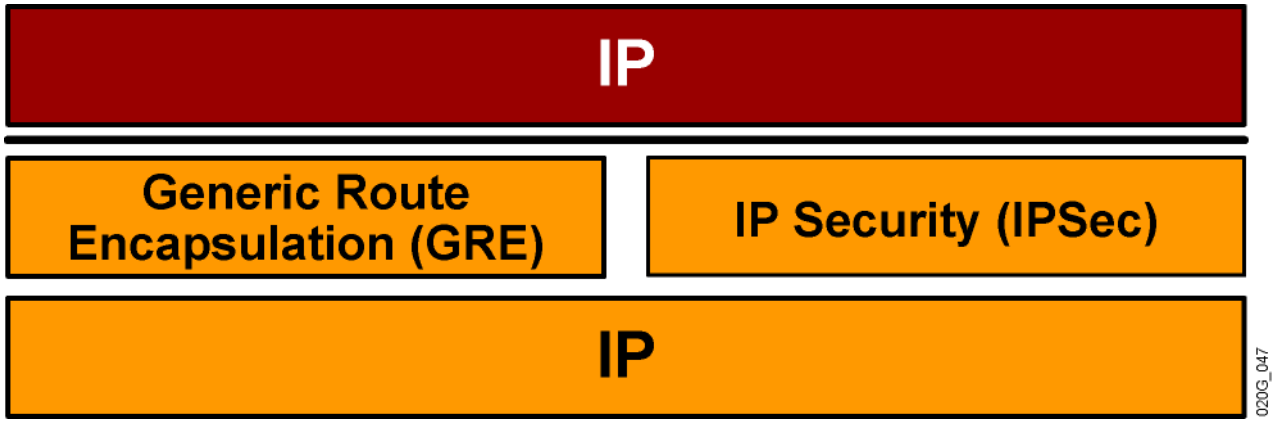
  Service provider establishes Layer 2 virtual circuits between customer sites.

  Customer is responsible for all higher layers.

Overlay VPNs (Cont.)
Frame Relay Example



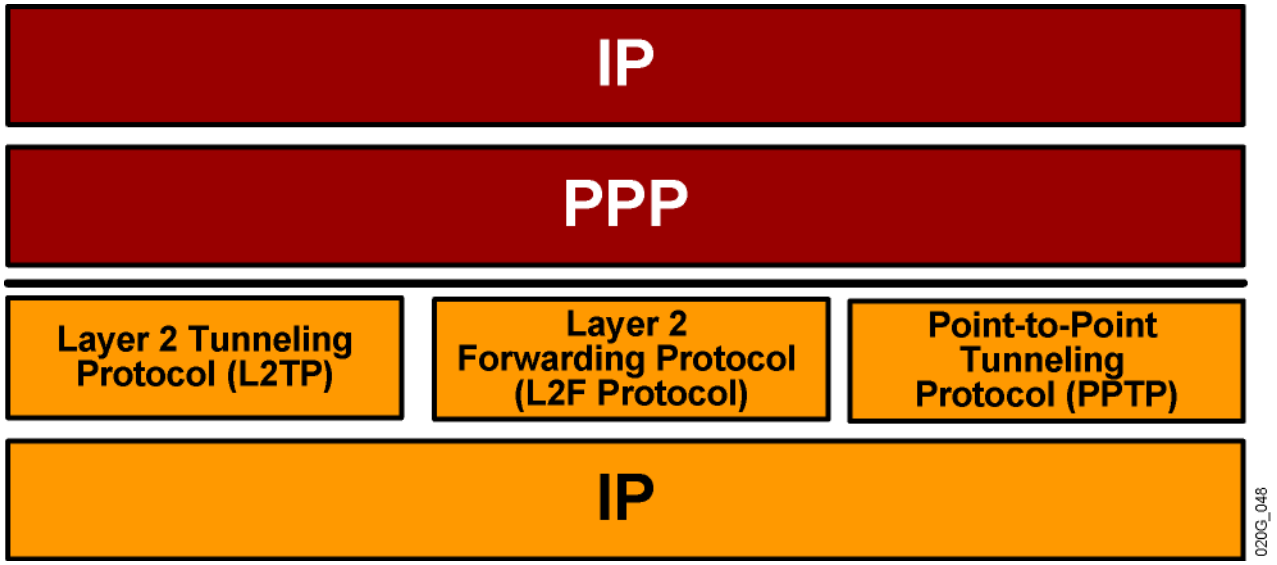
Overlay VPNs (Cont.)
IP Tunneling



VPN is implemented with IP-over-IP tunnels:

- Tunnels are established with GRE or IPSec.
- GRE is simpler (and quicker); IPSec provides authentication and security.

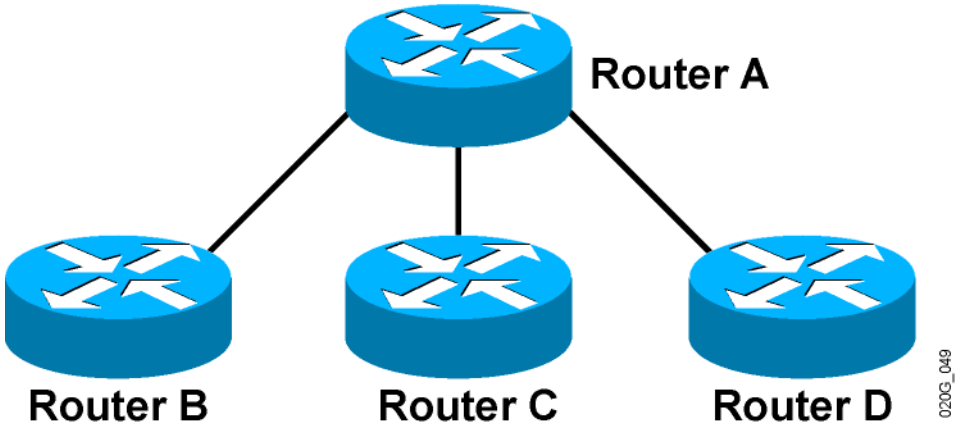
Overlay VPNs (Cont.)
Layer 2 Forwarding



VPN is implemented with PPP-over-IP tunnels.

Usually used in access environments (dialup, digital subscriber line).

Overlay VPNs (Cont.)
Layer 3 Routing

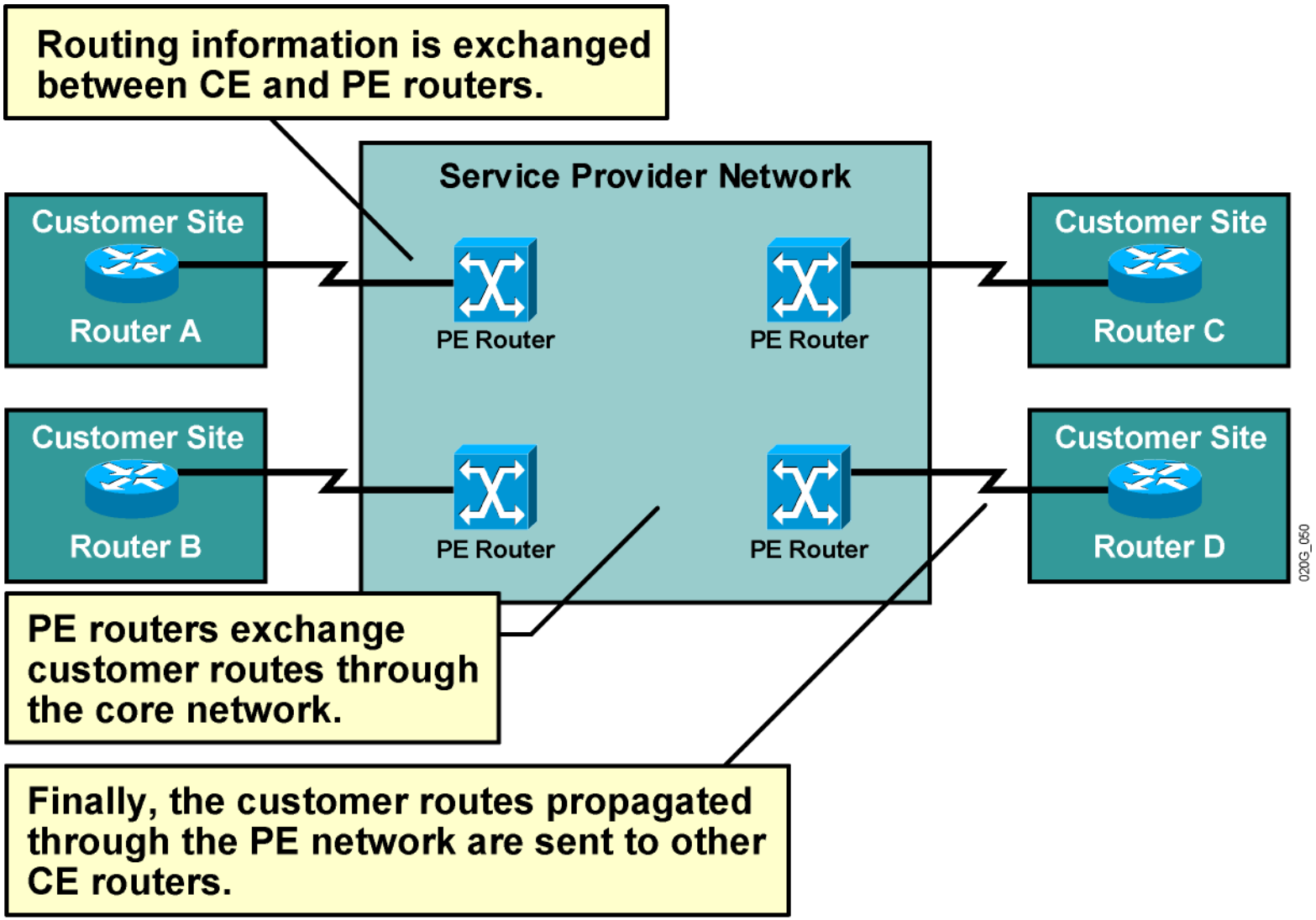


Service provider infrastructure appears as point-to-point links to customer routes.

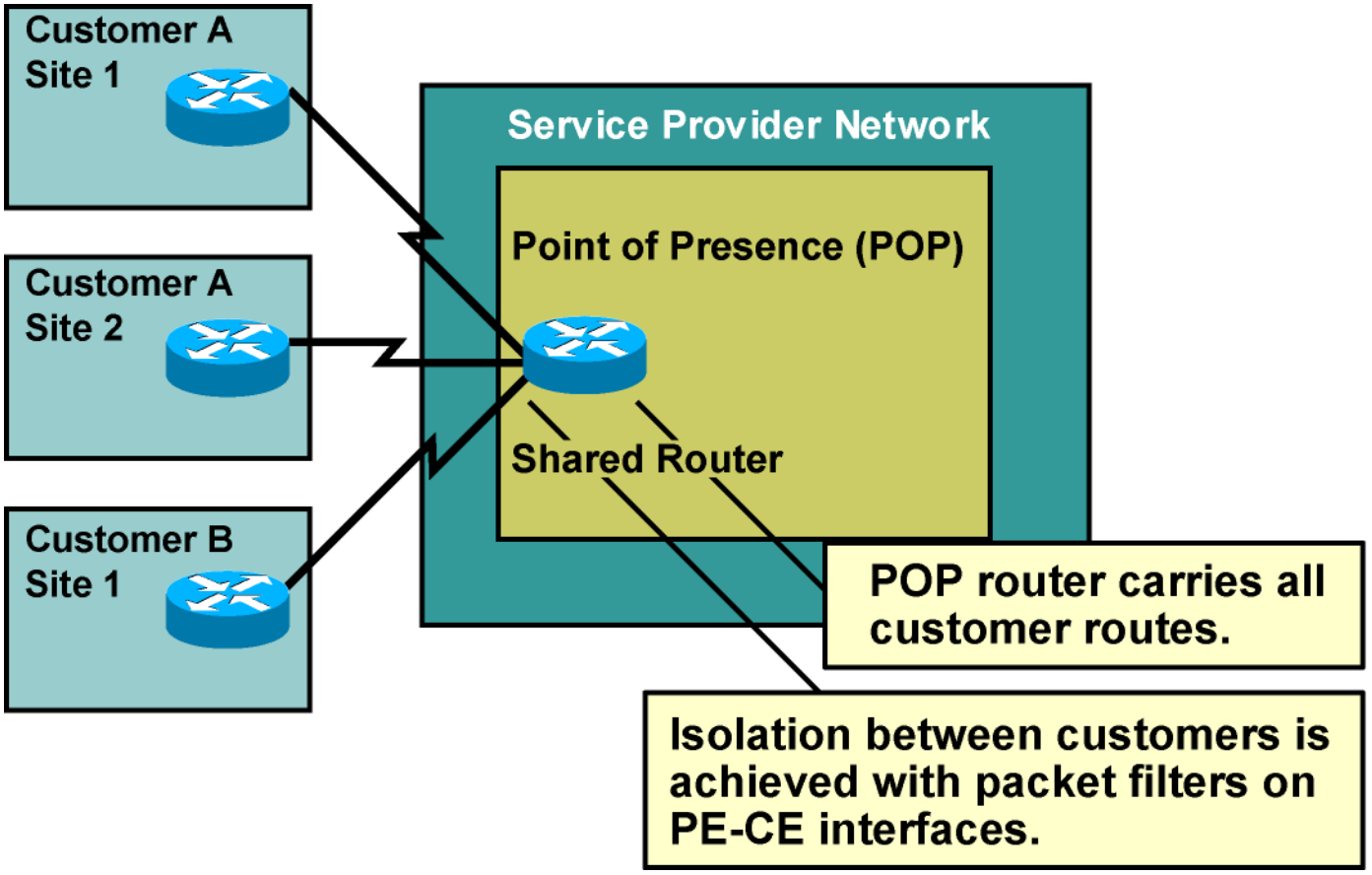
Routing protocols run directly between customer routers.

Service provider does not see customer routes and is responsible only for providing point-to-point transport of customer data.

Peer-to-Peer VPNs



Peer-to-Peer VPNs (Cont.)
Packet Filters



Benefits of VPN Implementations

Overlay VPN:

- Well-known and is easy to implement.
- Service provider does not participate in customer routing.
- Customer network and service provider network are well isolated.

Peer-to-peer VPN:

- Guarantees optimum routing between customer sites.
- Easier to provision an additional VPN.
- Only the sites are provisioned, not the links between them.

Drawbacks of VPN Implementations

Overlay VPN:

- Implementing optimum routing requires full mesh of virtual circuits.
- Virtual circuits have to be provisioned manually.
- Bandwidth must be provisioned on a site-to-site basis.
- Overlay VPNs always incur encapsulation overhead.

Peer-to-peer VPN:

- Service provider participates in customer routing.
- Service provider becomes responsible for customer convergence.
- PE routers carry all routes from all customers.
- Service provider needs detailed IP routing knowledge.

Drawbacks of Traditional Peer-to-Peer VPNs

Shared PE router:

- All customers share the same (provider-assigned or public) address space.
- High maintenance costs are associated with packet filters.
- Performance is lower—each packet has to pass a packet filter.

Dedicated PE router:

- All customers share the same address space.
- Each customer requires a dedicated router at each POP.

Summary

The two major VPN models are overlay and peer-to-peer.

Overlay VPNs can be implemented using Layer 1, Layer 2, and Layer 3 technologies.

Traditional peer-to-peer VPNs are implemented using IP routing technology.

Overlay VPNs use well-known technologies and are easy to implement, but require a full mesh of virtual circuits to provide optimum routing.

Summary

Peer-to-peer VPNs guarantee optimum routing between customer sites but require that the service provider participates in customer routing.

Both shared PE router and dedicated PE router implementations of peer-to-peer VPNs require the customers to share a common address space.

MPLS Bootcamp

VPN Types

Outline

Overview

VPN Categorization

Hub-and-Spoke Topology

Partial Mesh Overlay VPN

VPN Business Categorization

Extranet VPN

VPN Connectivity Categorization

Central Services Extranet

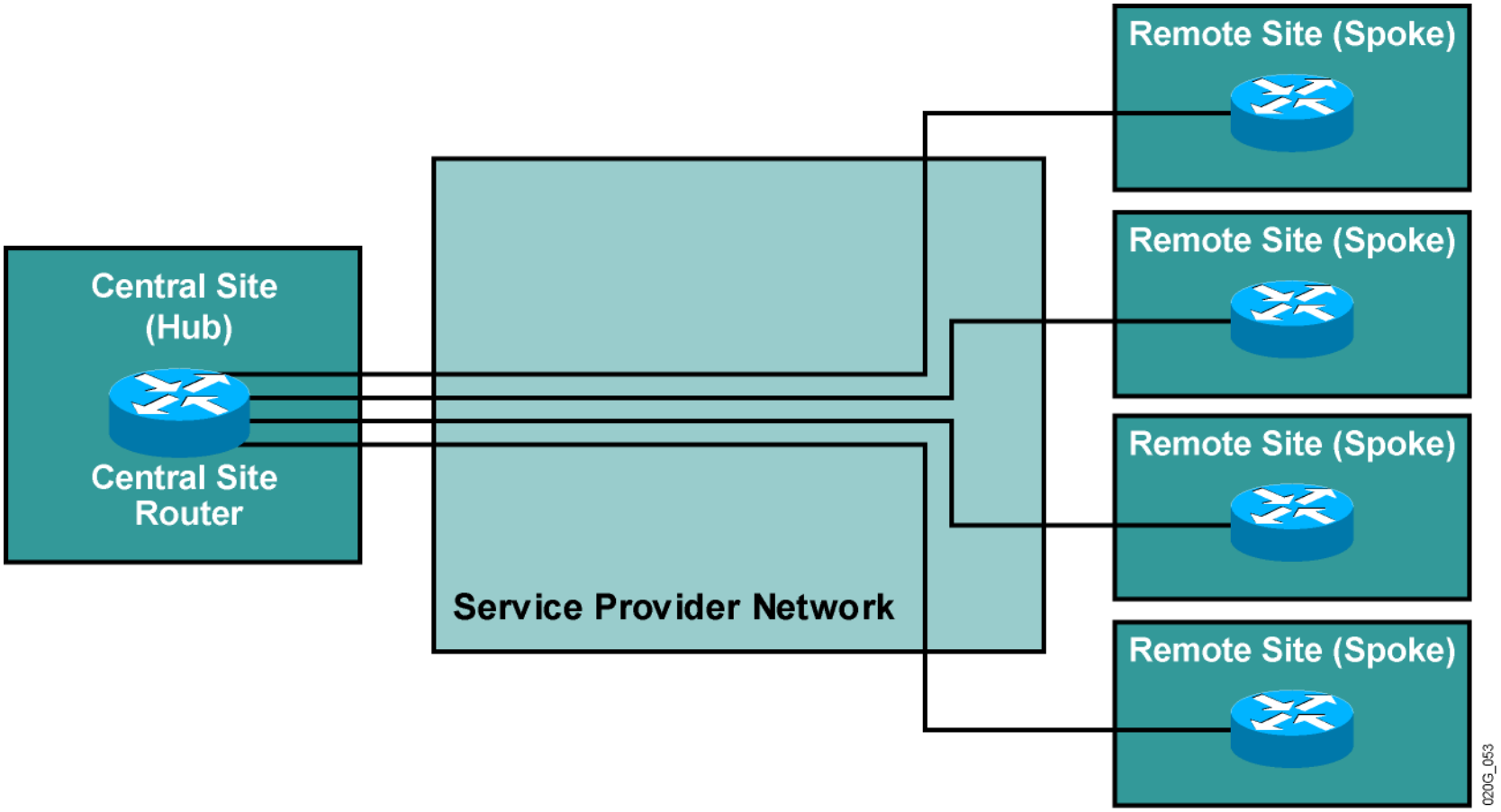
Managed Network Overlay VPN Implementation

Lesson Summary

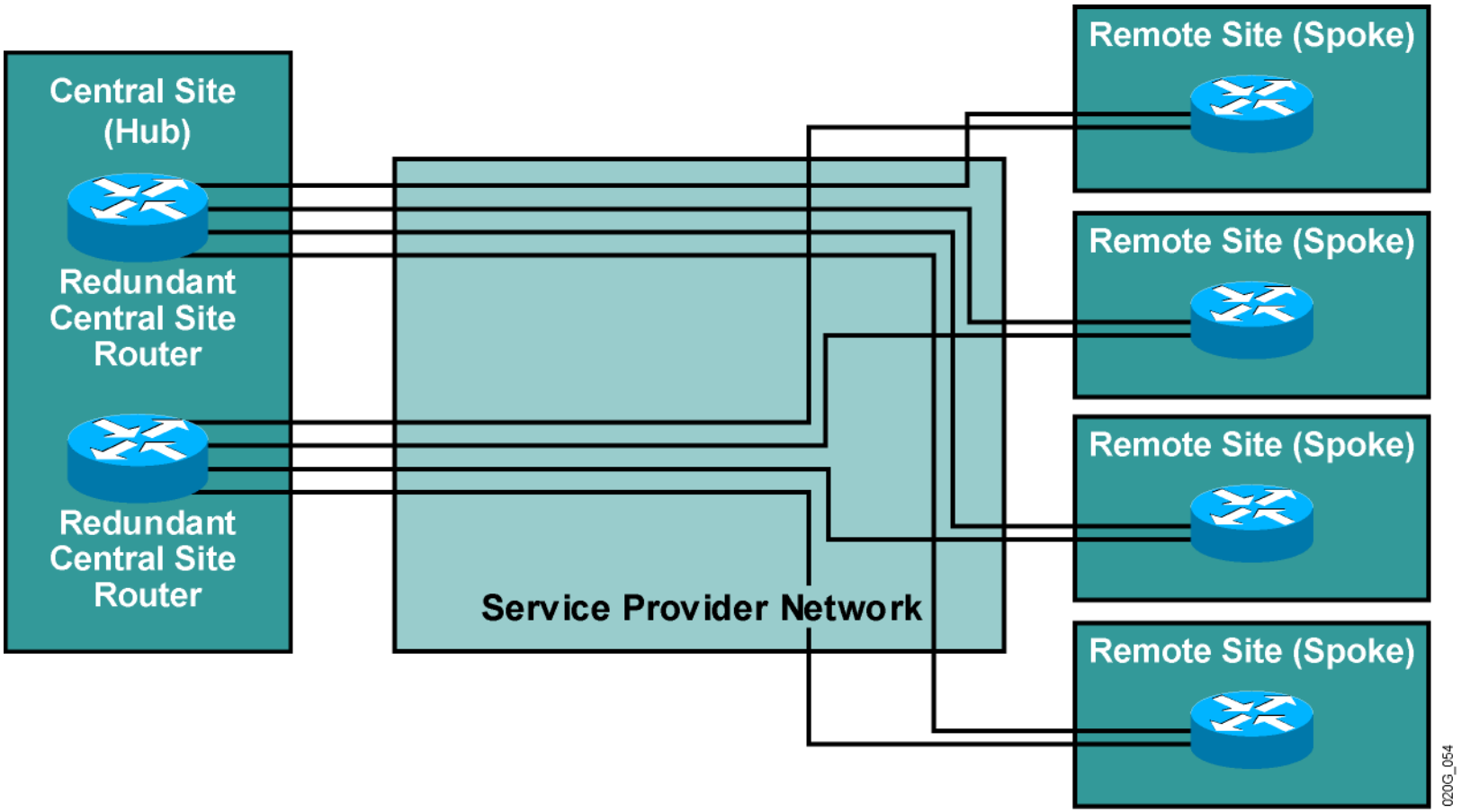
Overlay VPN Topology Category

- Overlay VPNs are categorized based on the topology of the virtual circuits:
  - (Redundant) hub-and-spoke
  - Partial mesh
  - Full mesh
  - Multilevel—combines several levels of overlay VPN topologies

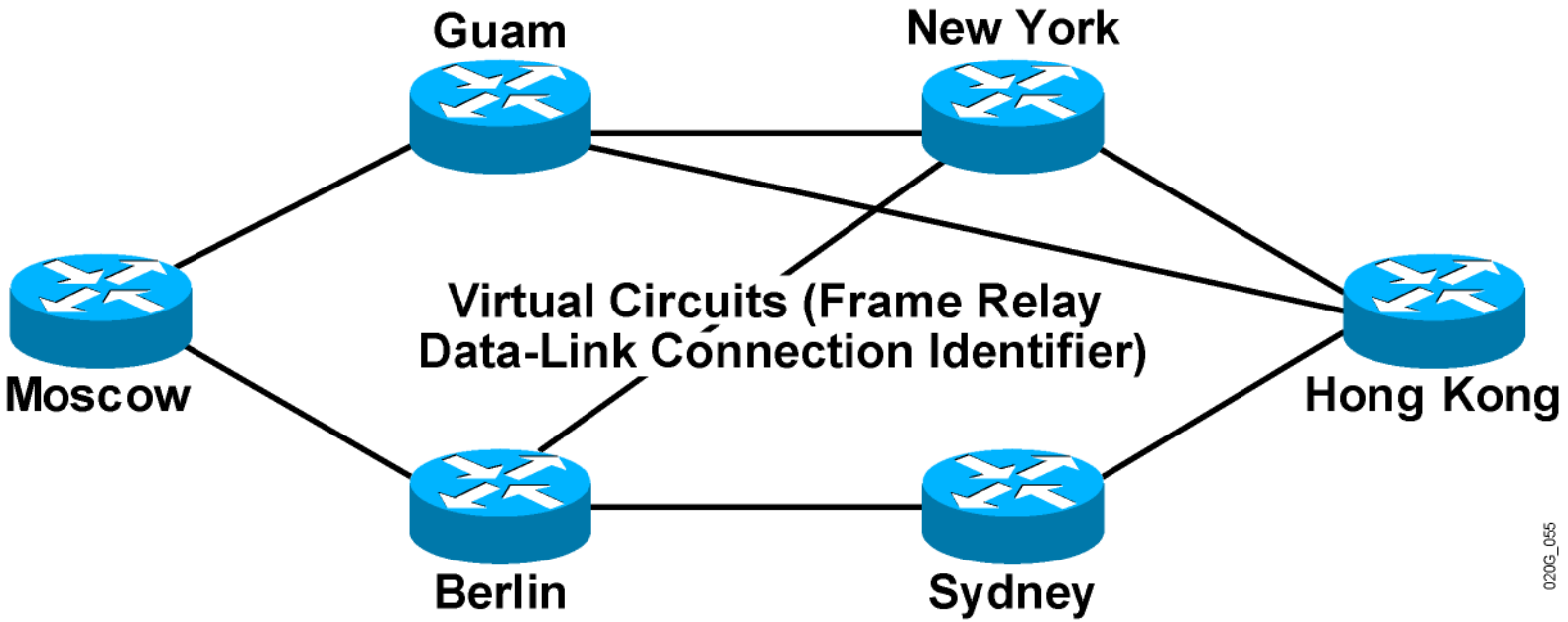
Hub-and-Spoke Overlay VPN Topology



Hub-and-Spoke Overlay VPN Topology (Cont.)
Redundant Hub-and-Spoke Topology



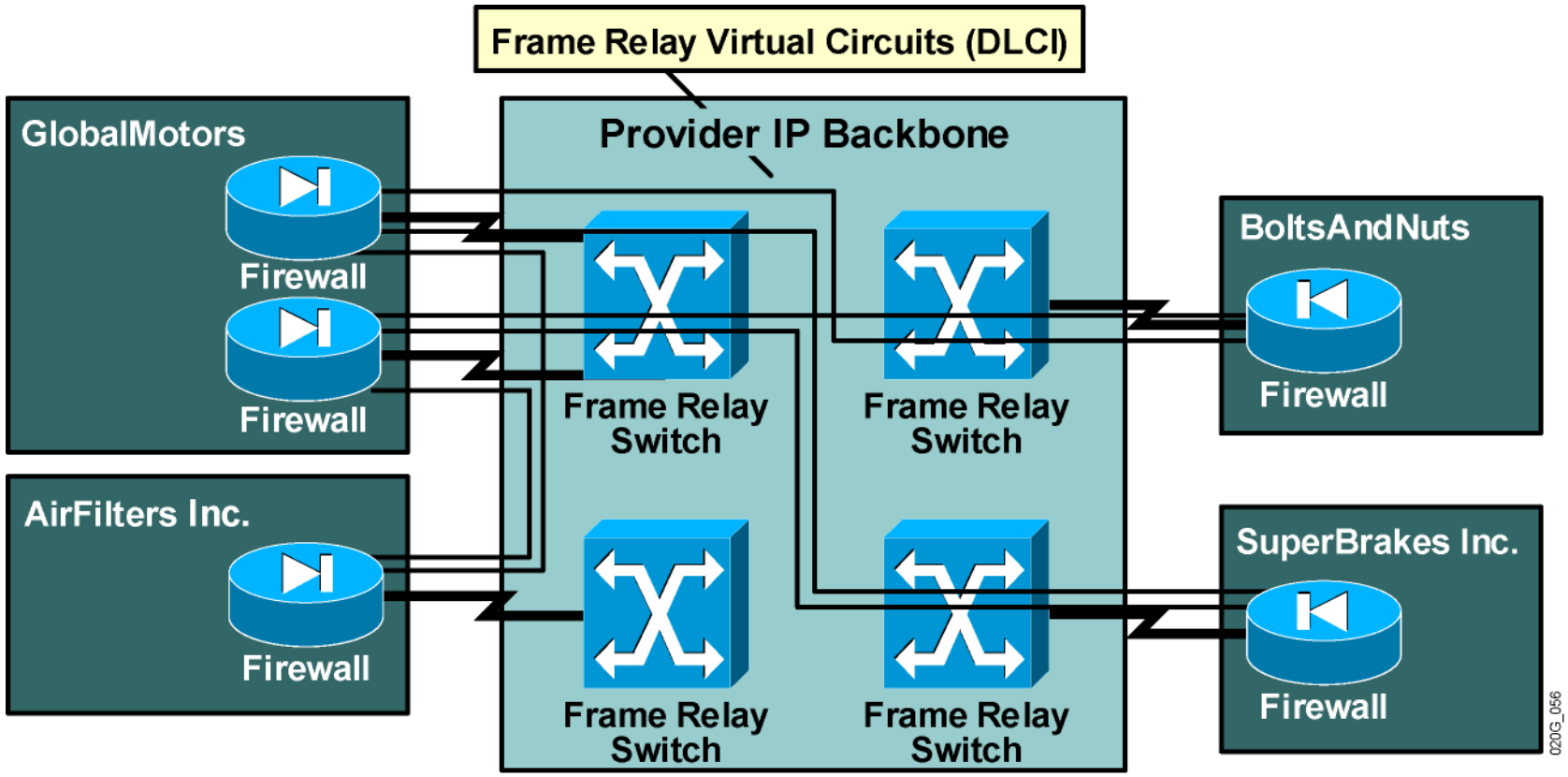
Partial Mesh Overlay VPN Topology



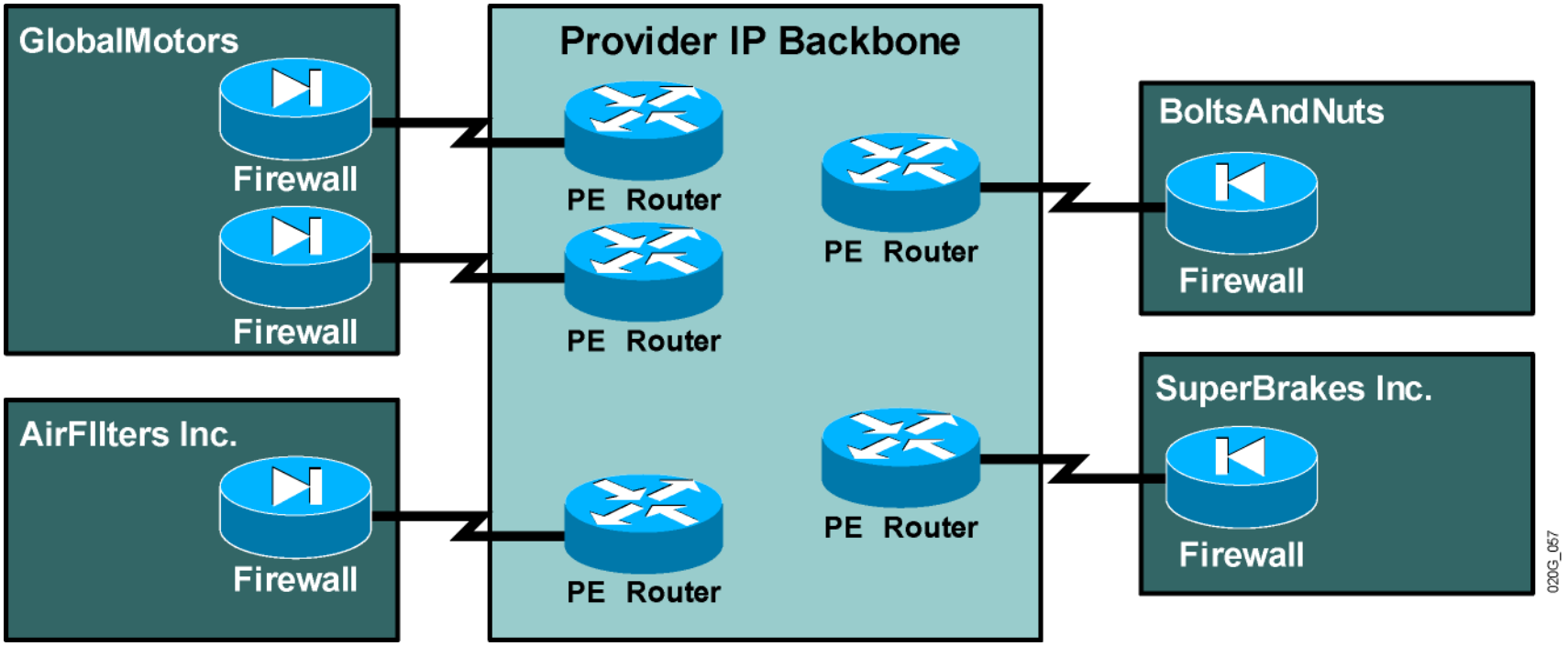
VPN Business Category

- VPNs can be categorized on the business needs that they fulfill:
  - Intranet VPN: Connects sites within an organization.
  - Extranet VPN: Connects different organizations in a secure way.
  - Access VPN: VPDN provides dialup access into a customer network.

Extranet VPNs
Overlay VPN Implementation



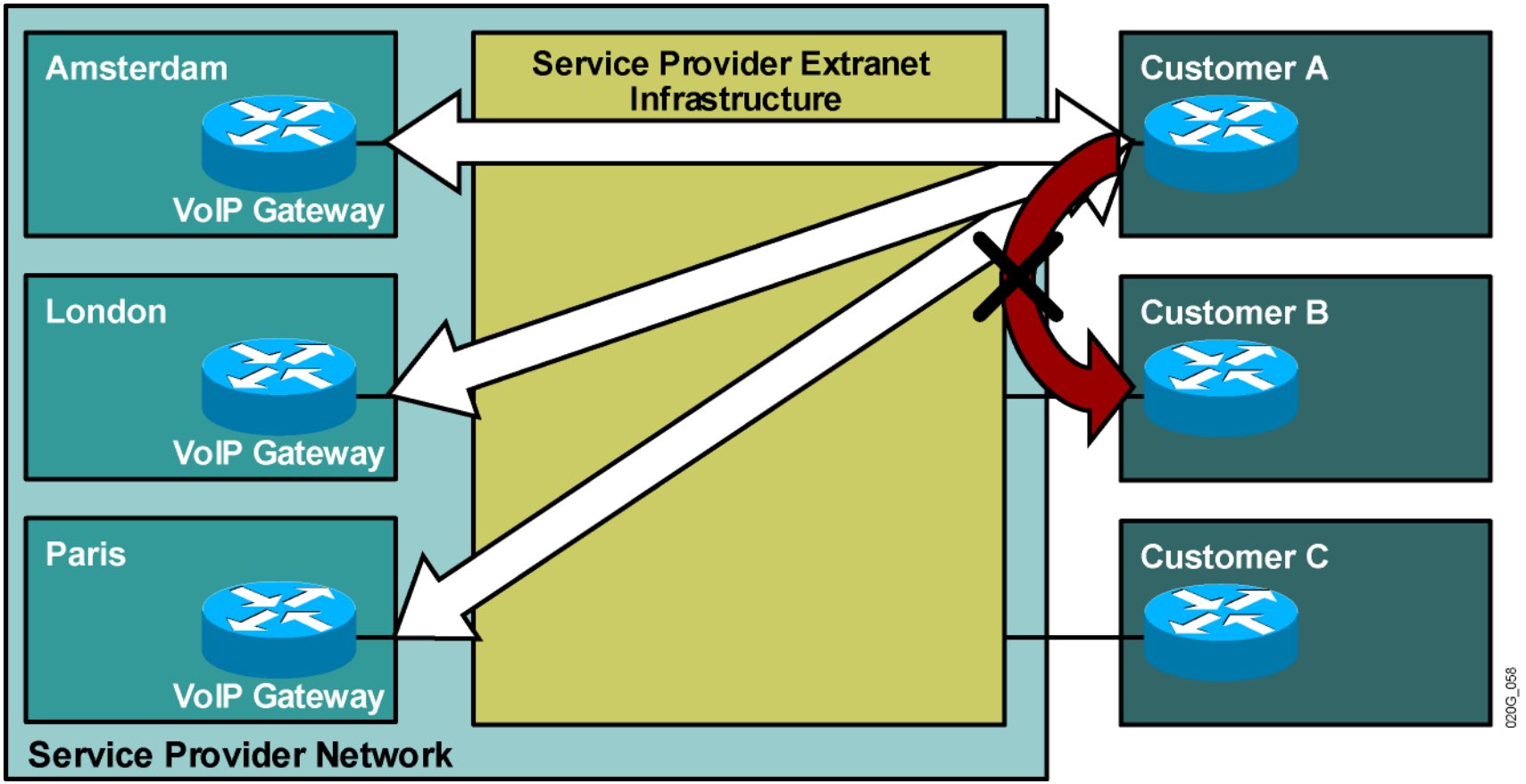
Extranet VPNs (Cont.)
Peer-to-Peer VPN Implementation



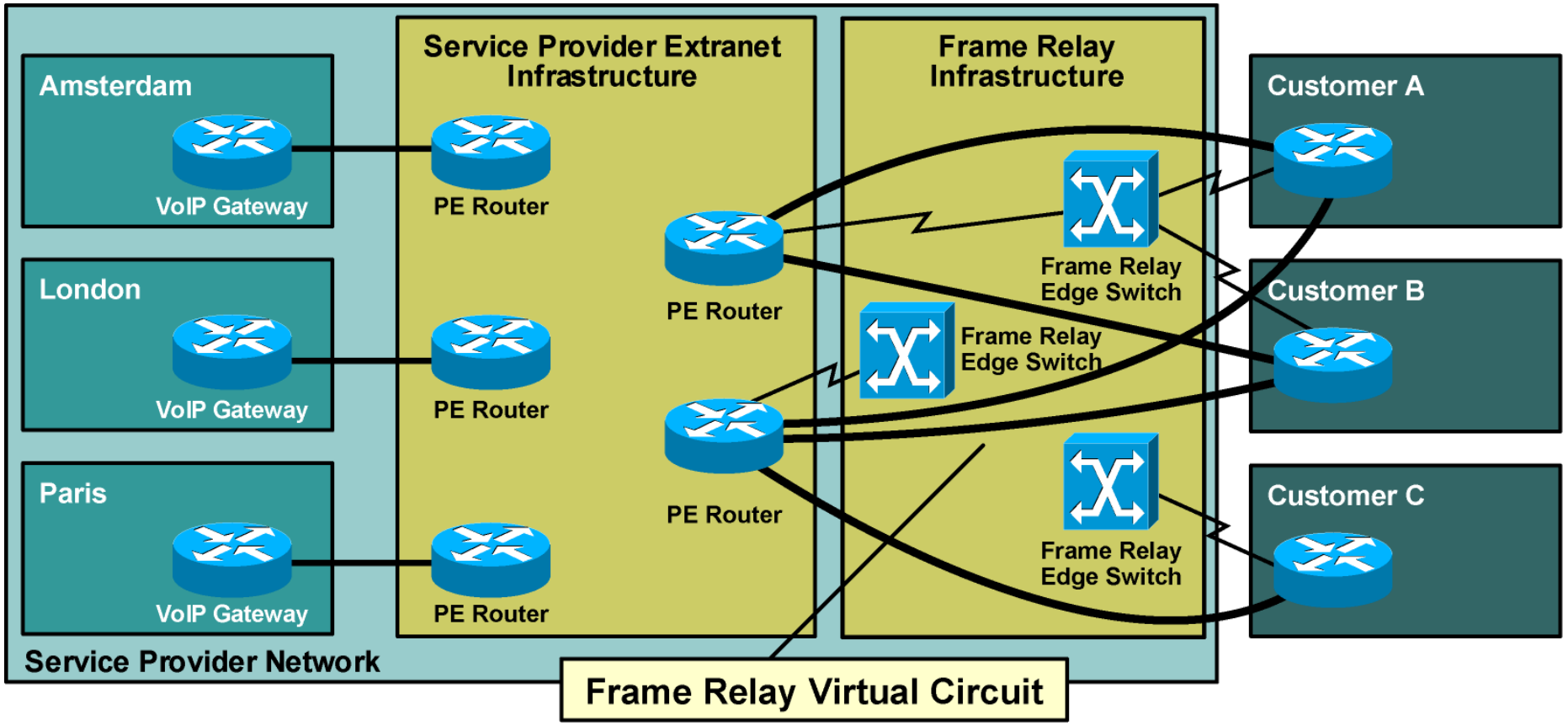
VPN Connectivity Category

- VPNs can also be categorized according to the connectivity required between sites:
  - Simple VPN: Every site can communicate with every other site.
  - Overlapping VPN: Some sites participate in more than one simple VPN.
  - Central services VPN: All sites can communicate with central servers but not with each other.
  - Managed network: A dedicated VPN is established to manage CE routers.

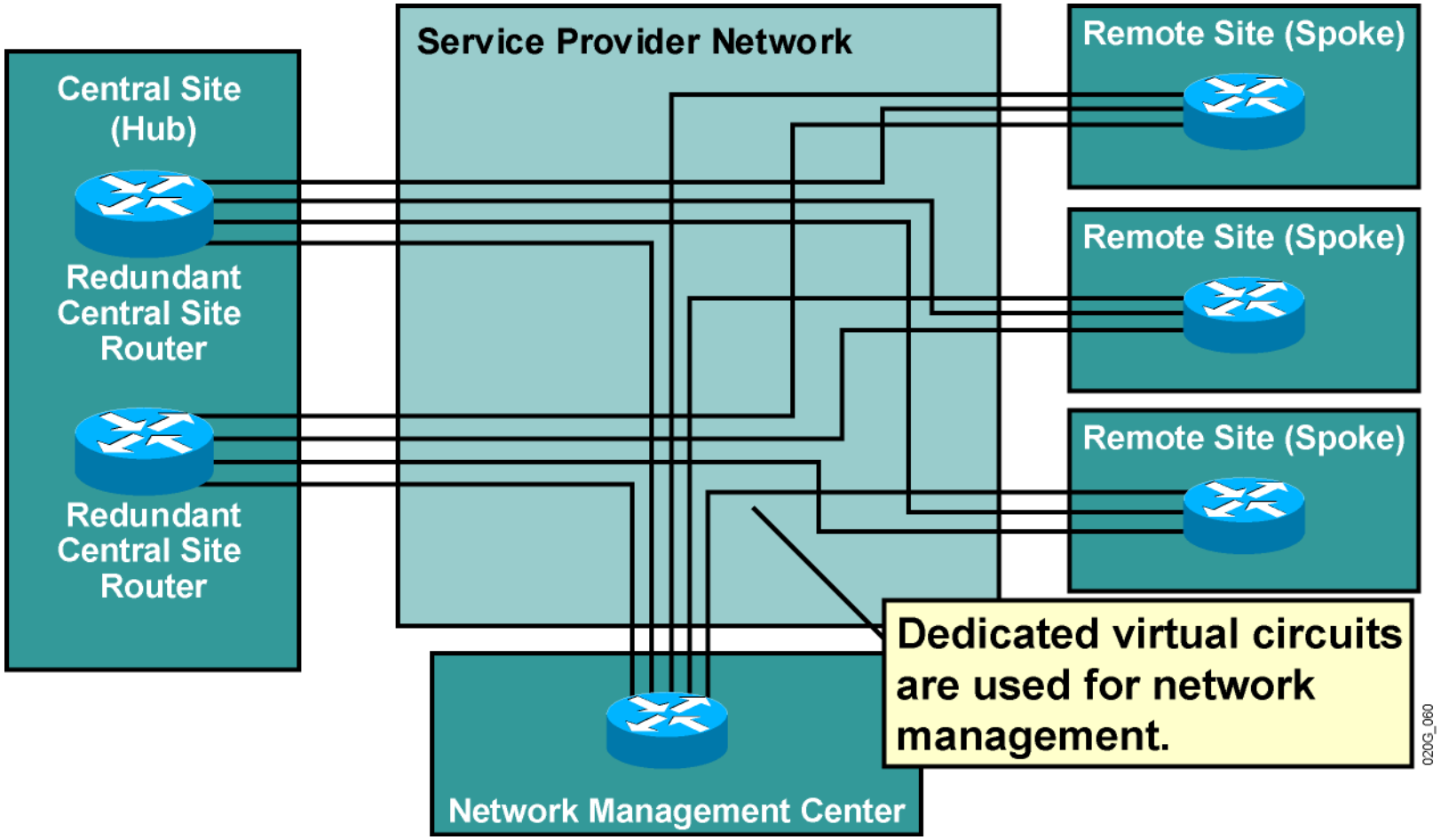
Central Services Extranet



Central Services Extranet (Cont.)
Hybrid (Overlay + Peer-to-Peer) Implementation



Managed Network Overlay VPN Implementation



Summary

Major VPN topologies consist of the following:

- Hub-and-spoke – simplest topology
- Partial mesh – cost/complexity factors dictate
- Full mesh – connections between all sites
- Multilevel – can be used for large-scale networks

VPNs can be based on business needs:

- Intranet
- Extranet
- Access

MPLS Bootcamp

MPLS VPN Architecture

Outline

Overview

MPLS VPN Architecture

PE Router Architecture

Propagation Routing Information across the P-network

Route Distinguishers

Route Targets

Virtual Private Networks Redefined

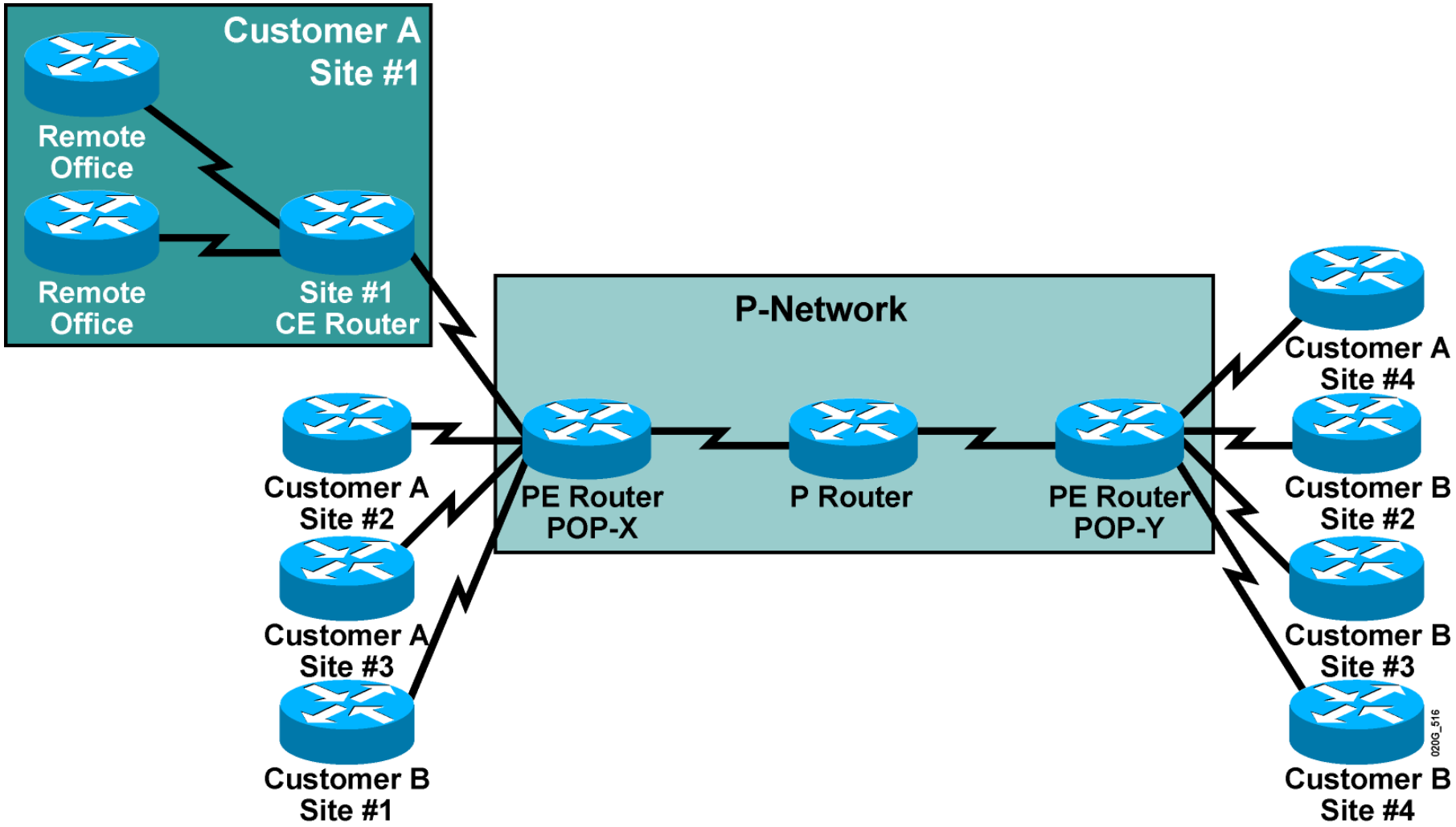
Impact of Complex VPN Topologies on Virtual Routing Tables

Lesson Summary

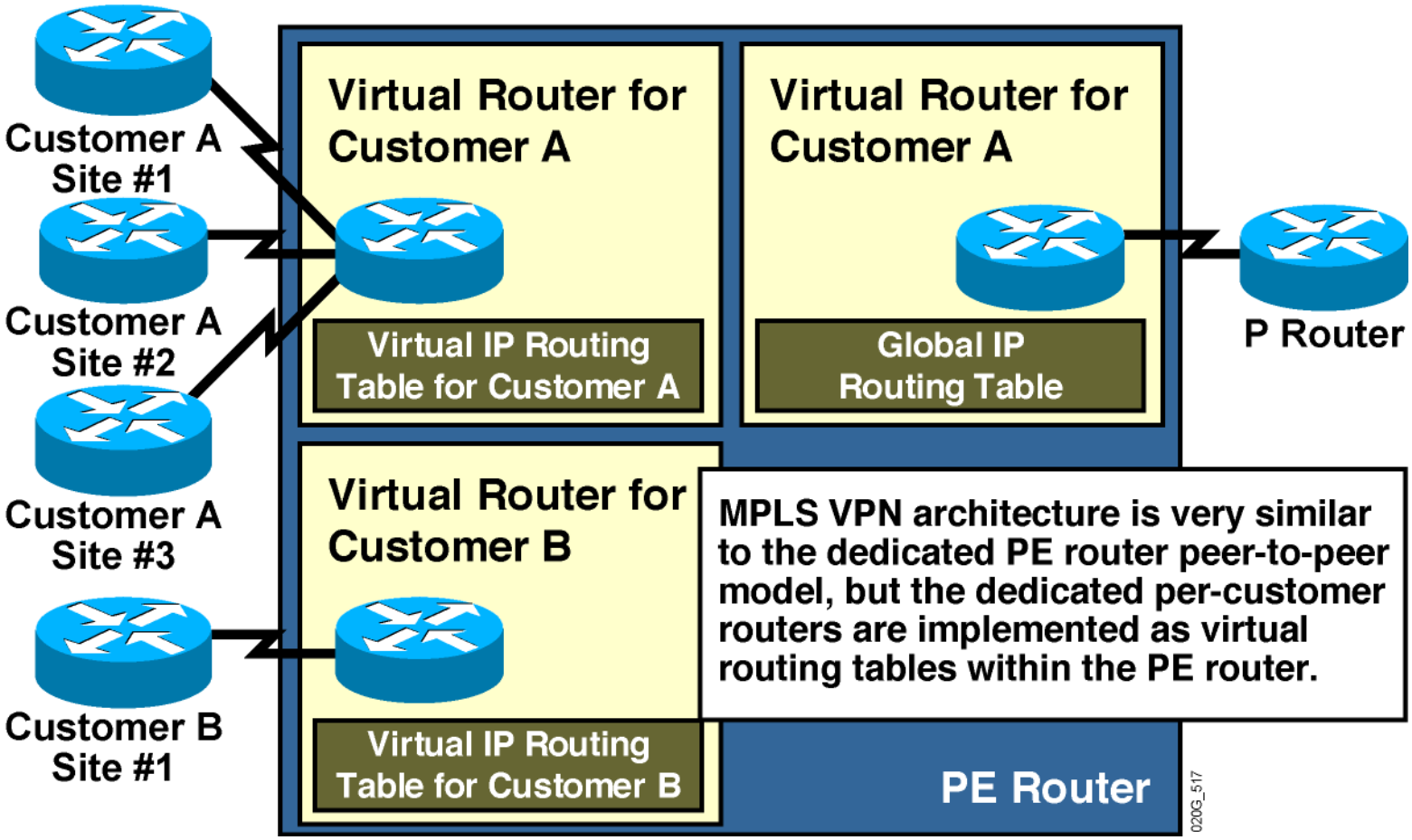
MPLS VPN Architecture

- An MPLS VPN combines the best features of an overlay VPN and a peer-to-peer VPN:
  - PE routers participate in customer routing, guaranteeing optimum routing between sites and easy provisioning.
  - PE routers carry a separate set of routes for each customer (similar to the dedicated PE router approach).
  - Customers can use overlapping addresses.

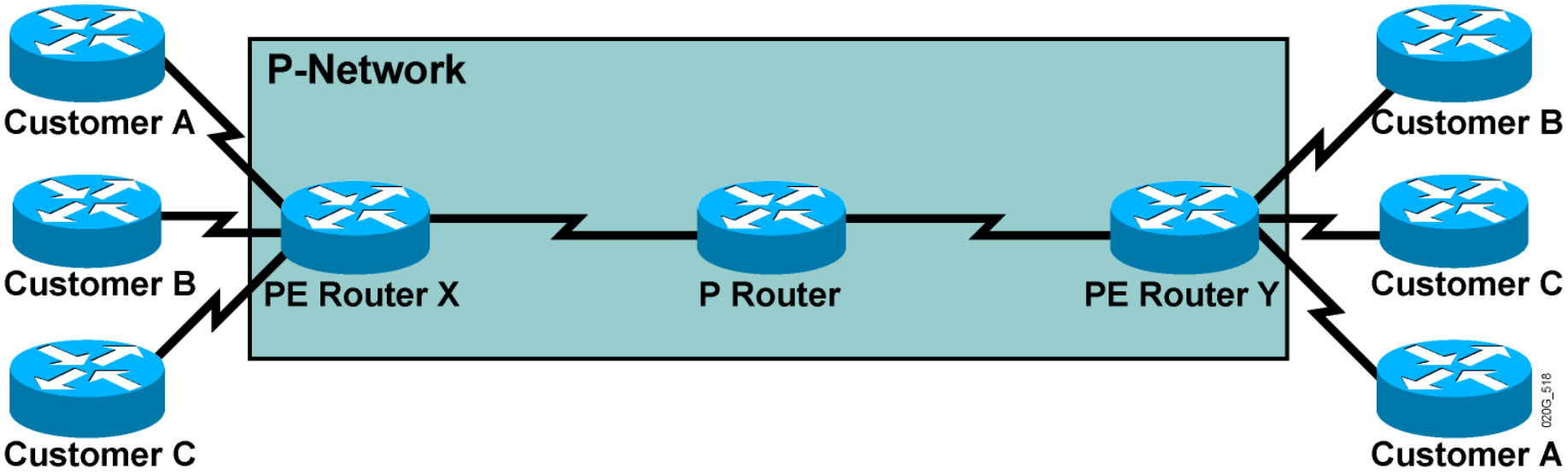
MPLS VPN Architecture (Cont.) Terminology



PE Router Architecture

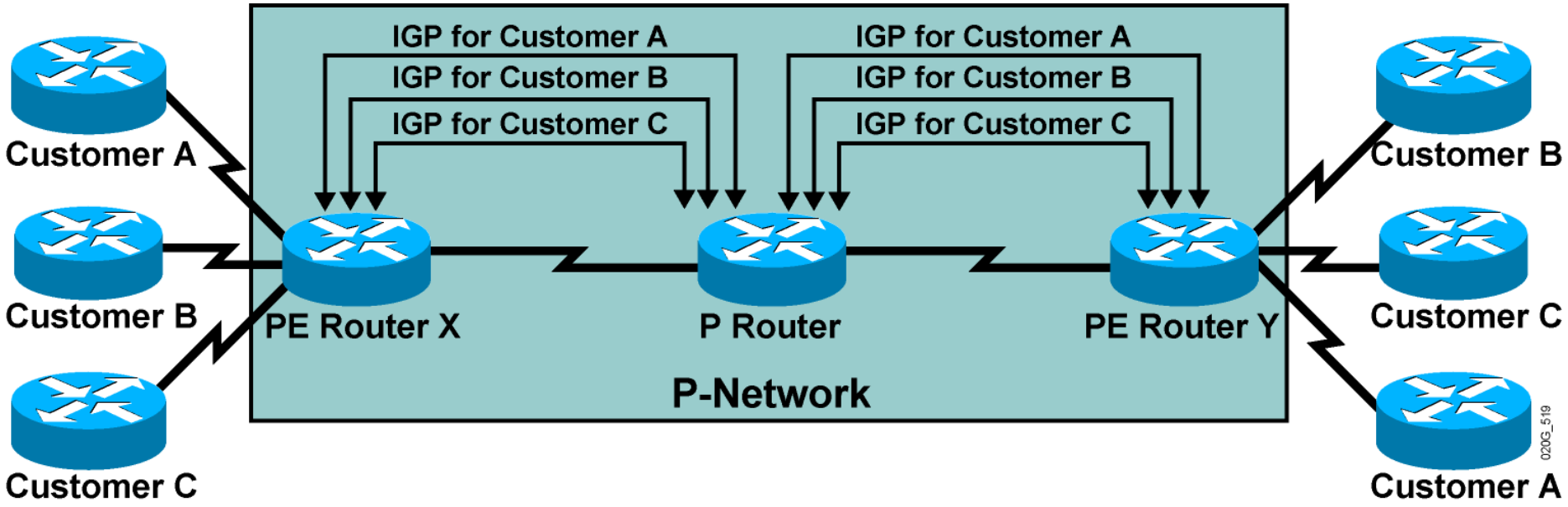


Propagation of Routing Information Across the P-Network



Question: How will PE routers exchange customer routing information?

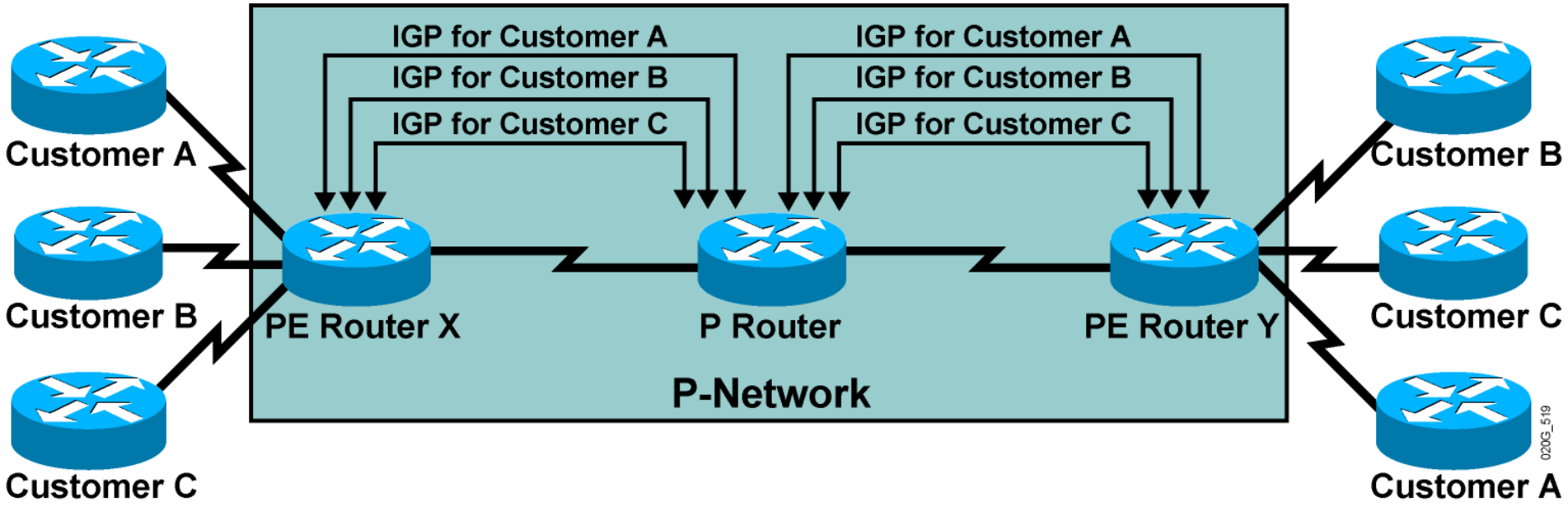
Propagation of Routing Information Across the P-Network



Question: How will PE routers exchange customer routing information?

Answer #1: Run a dedicated Interior Gateway Protocol (IGP) for each customer across the P-network.

Propagation of Routing Information Across the P-Network



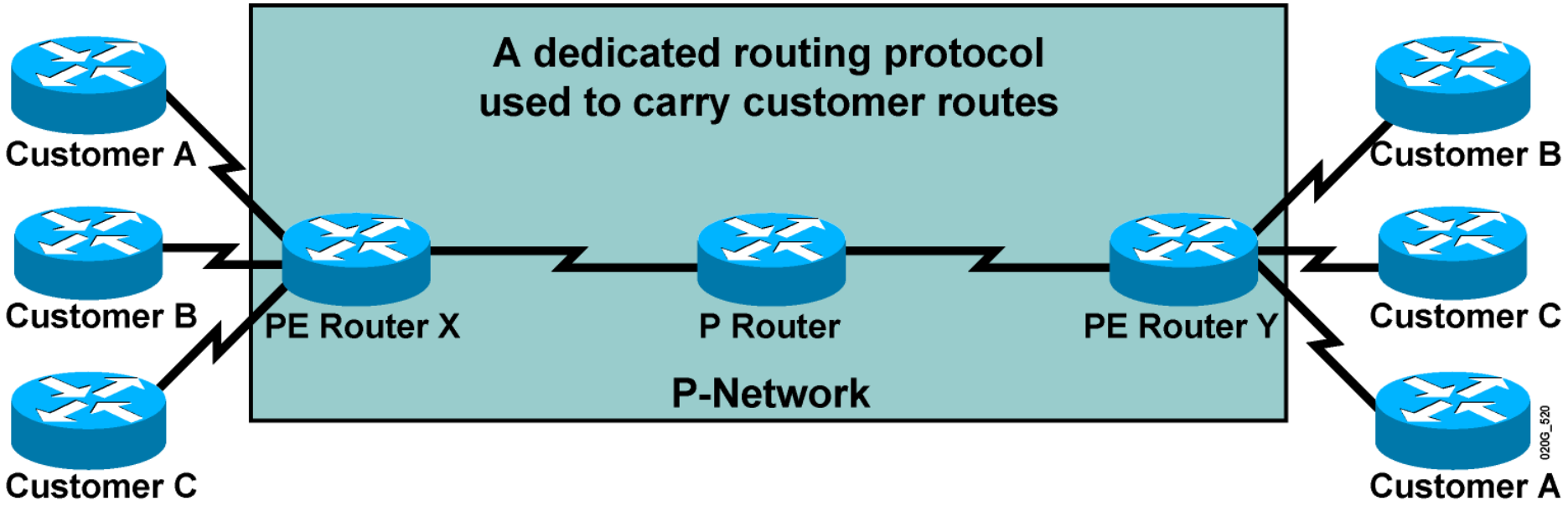
Question: How will PE routers exchange customer routing information?

Answer #1: Run a dedicated Interior Gateway Protocol (IGP) for each customer across the P-network.

This is the wrong answer for the following reasons:

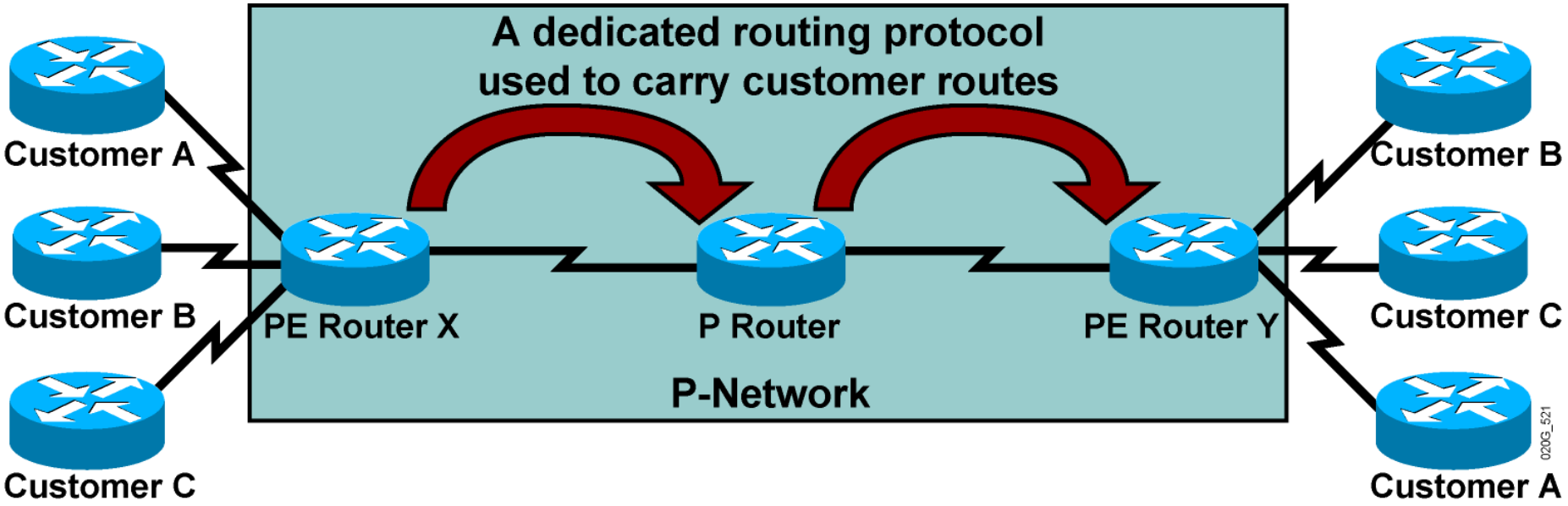
- The solution does not scale.
- P routers carry all customer routes.

Propagation of Routing Information Across the P-Network (Cont.)



Question: How will PE routers exchange customer routing information?

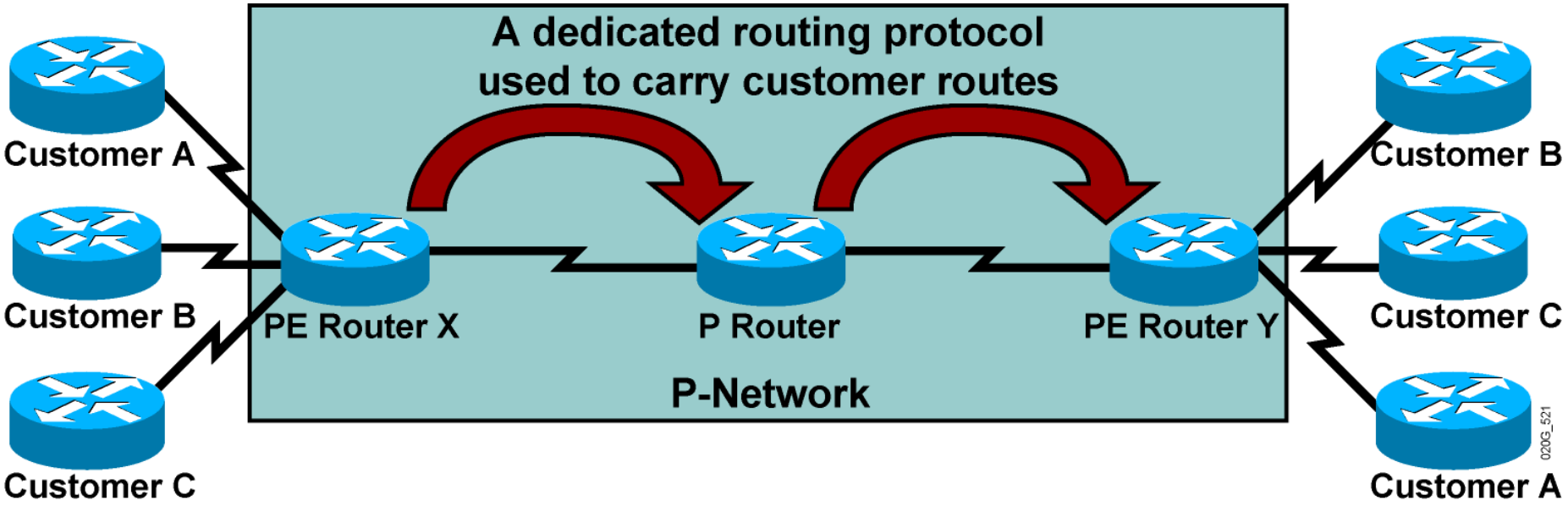
Propagation of Routing Information Across the P-Network (Cont.)



Question: How will PE routers exchange customer routing information?

Answer #2: Run a single routing protocol that will carry all customer routes inside the provider backbone.

Propagation of Routing Information Across the P-Network (Cont.)



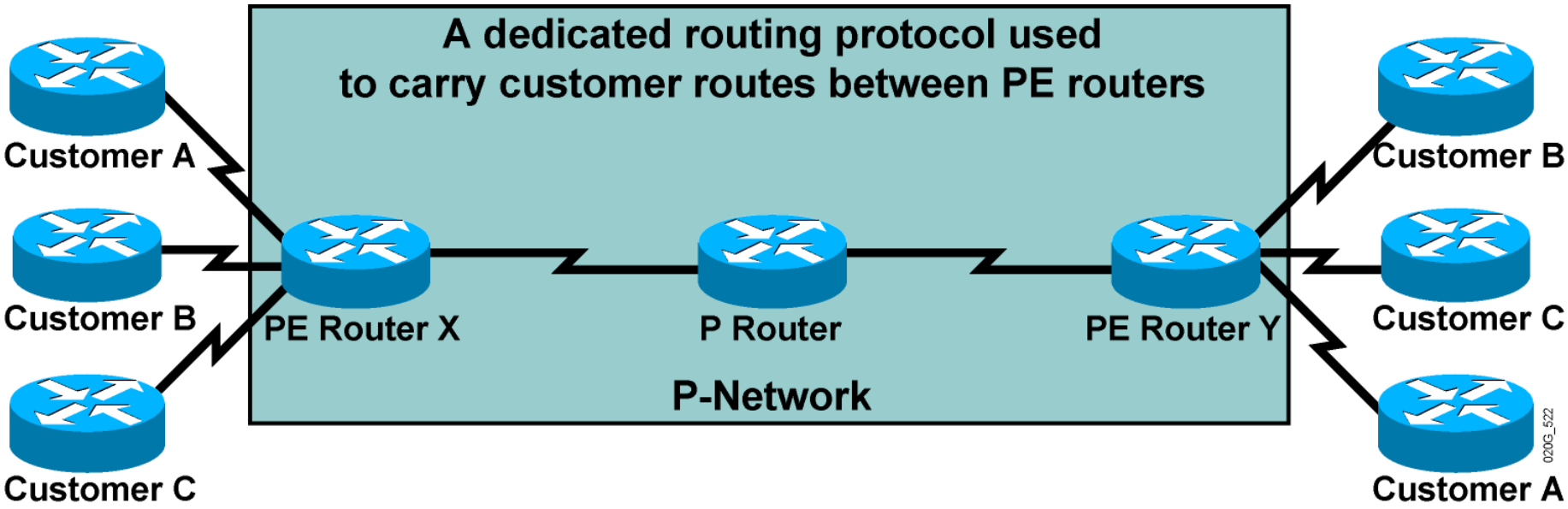
Question: How will PE routers exchange customer routing information?

Answer #2: Run a single routing protocol that will carry all customer routes inside the provider backbone.

Better answer, but still not good enough:

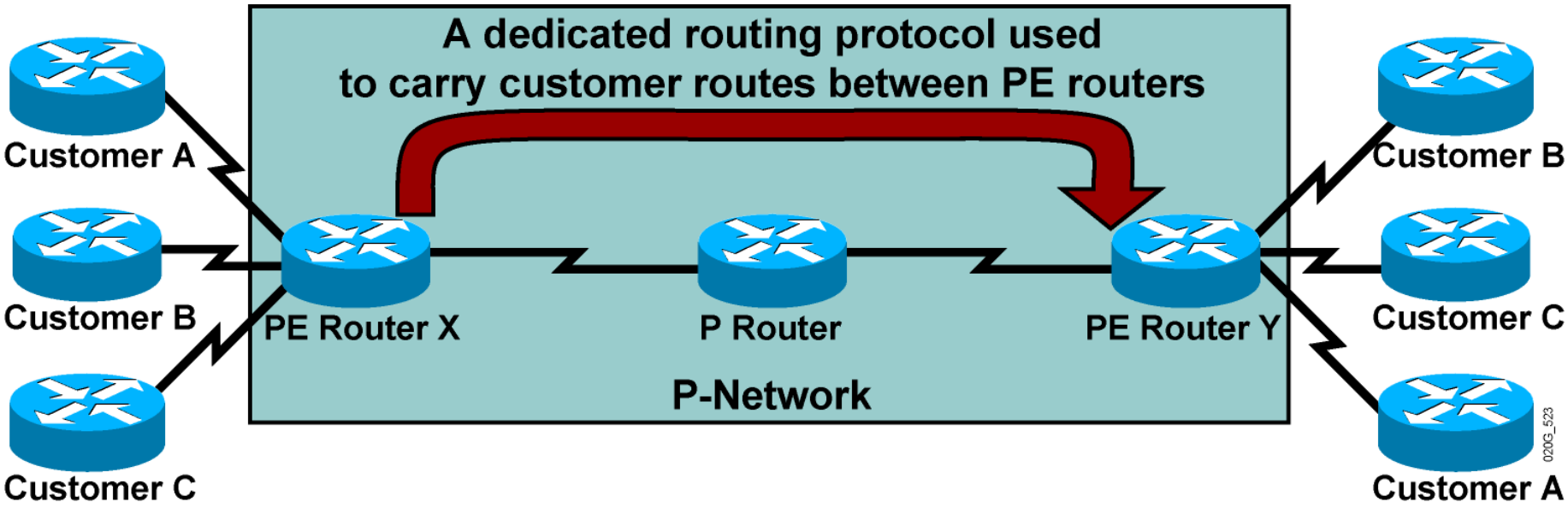
- P routers carry all customer routes.

Propagation of Routing Information Across the P-Network (Cont.)



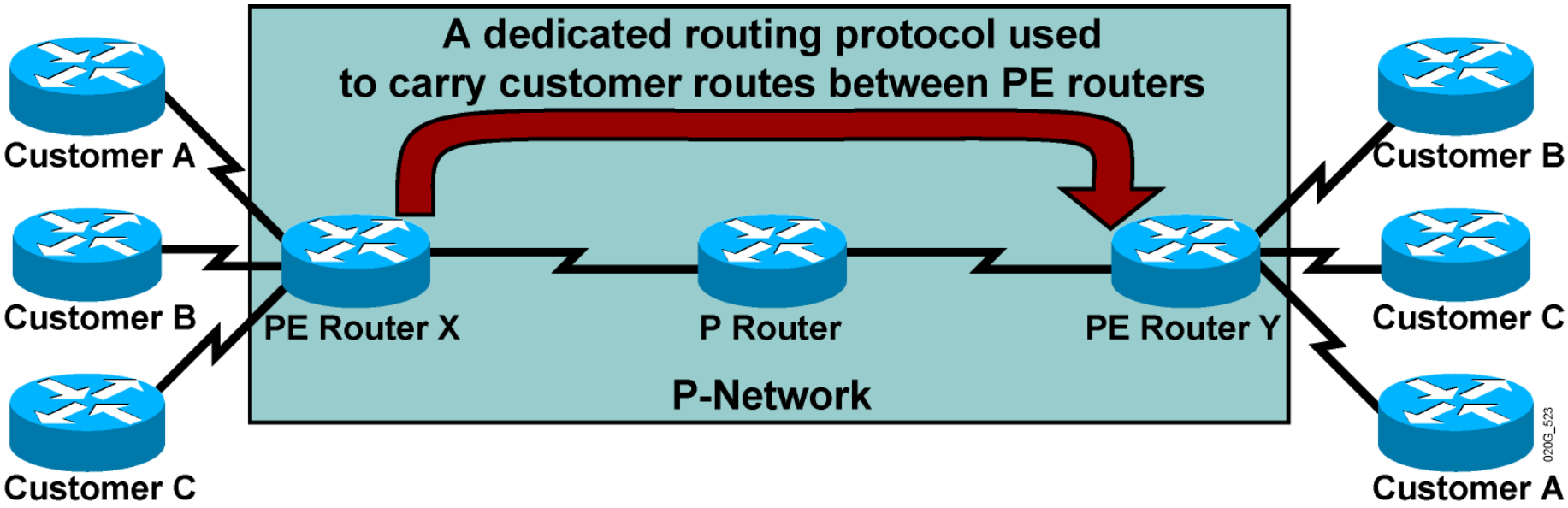
- Question: How will PE routers exchange customer routing information?

Propagation of Routing Information Across the P-Network (Cont.)



- Question: How will PE routers exchange customer routing information?
- Answer #3: Run a single routing protocol that will carry all customer routes between PE routers. Use MPLS labels to exchange packets between PE routers.

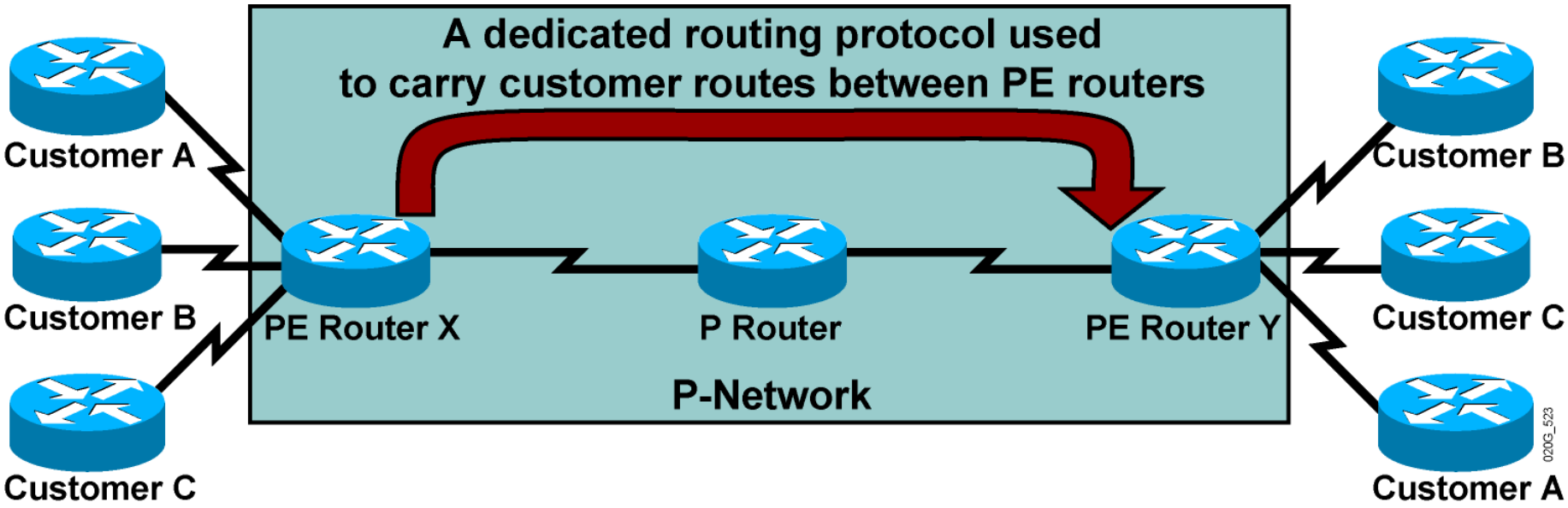
Propagation of Routing Information Across the P-Network (Cont.)



- Question: How will PE routers exchange customer routing information?
- Answer #3: Run a single routing protocol that will carry all customer routes between PE routers. Use MPLS labels to exchange packets between PE routers.
- The best answer:

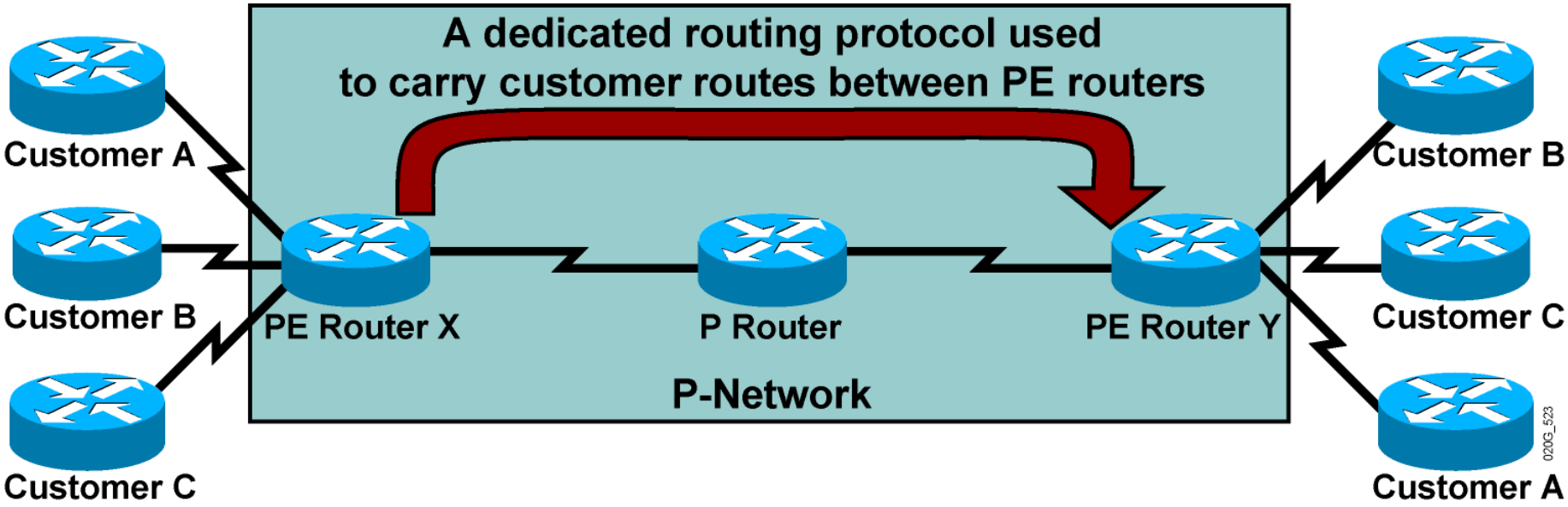
  P routers do not carry customer routes; the solution is scalable.

Propagation Routing Information Across the P-Network (Cont.)



Question: **Which protocol can be used to carry customer routes between PE routers?**

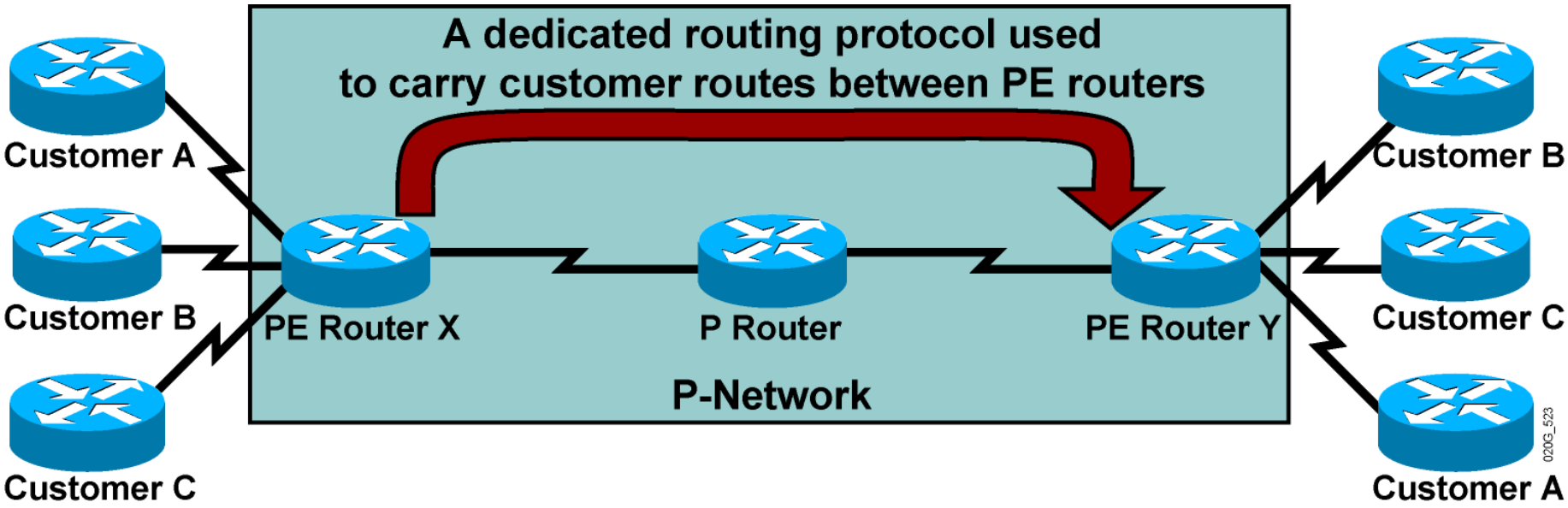
Propagation Routing Information Across the P-Network (Cont.)



Question: Which protocol can be used to carry customer routes between PE routers?

Answer: The number of customer routes can be very large. BGP is the only routing protocol that can scale to a very large number of routes.

Propagation Routing Information Across the P-Network (Cont.)



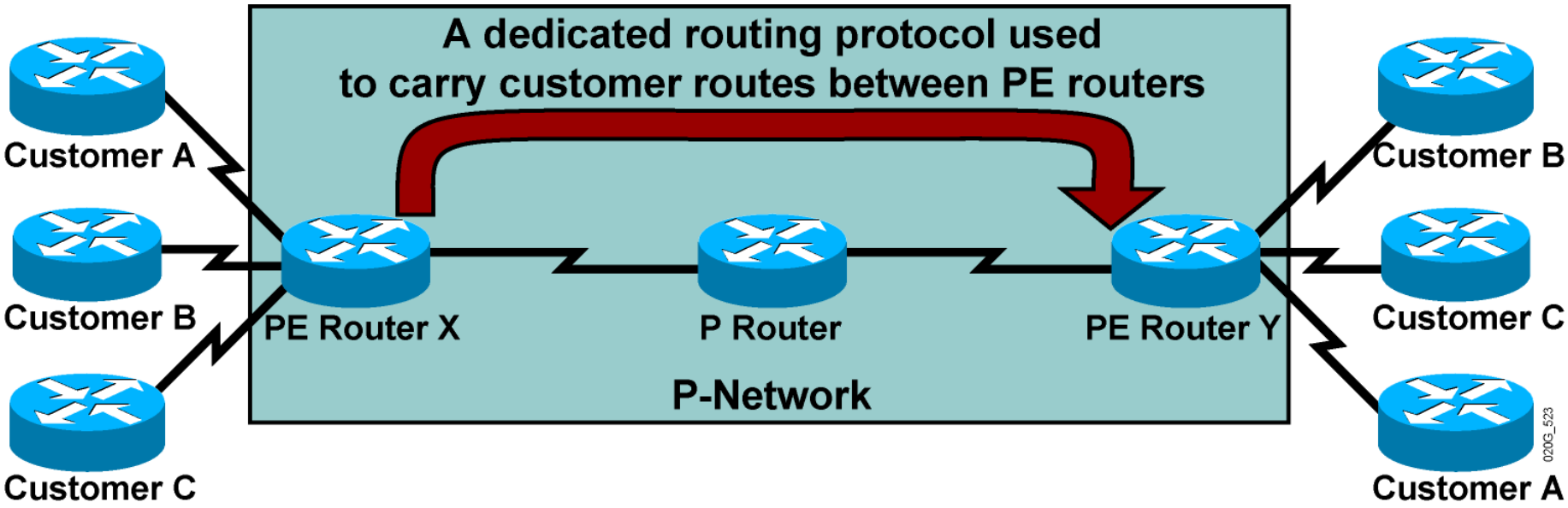
Question: Which protocol can be used to carry customer routes between PE routers?

Answer: The number of customer routes can be very large. BGP is the only routing protocol that can scale to a very large number of routes.

Conclusion:

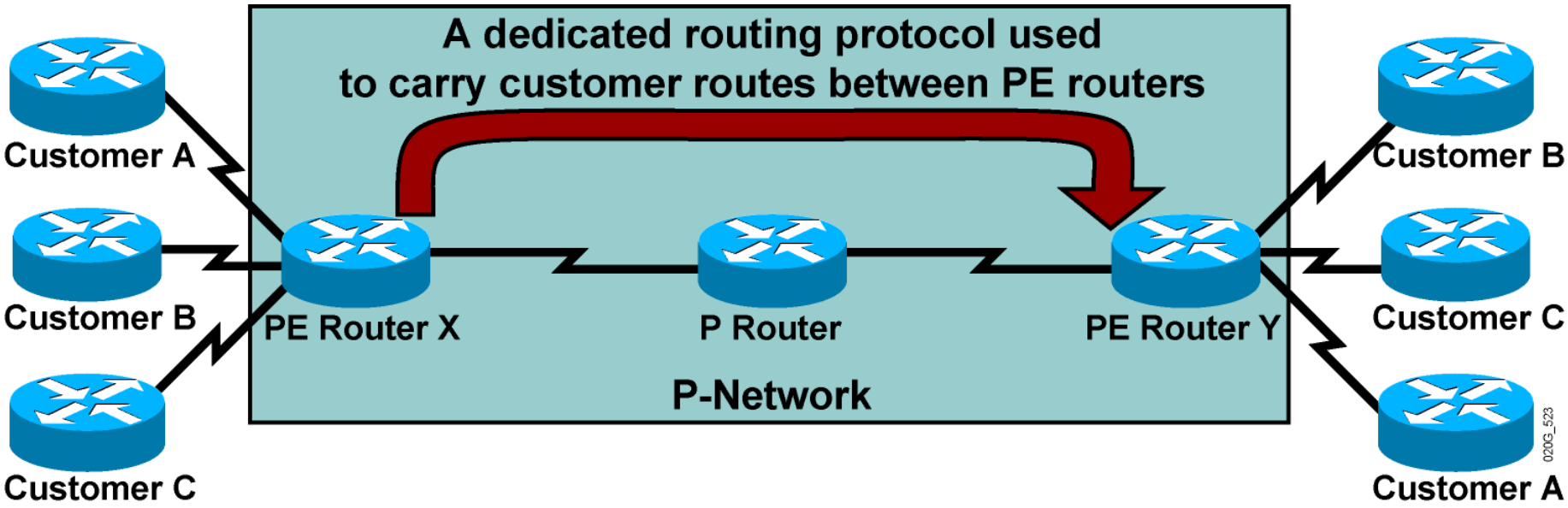
BGP is used to exchange customer routes directly between PE routers.

Propagation of Routing Information Across the P-Network (Cont.)



Question: How will information about the overlapping subnets of two customers be propagated via a single routing protocol?

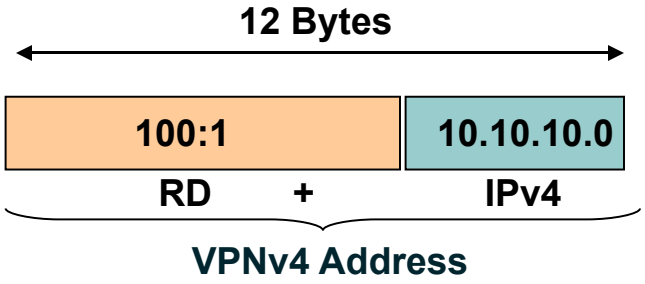
Propagation of Routing Information Across the P-Network (Cont.)



Question: How will information about the overlapping subnets of two customers be propagated via a single routing protocol?

Answer: Extend the customer addresses to make them unique.

Route Distinguishers



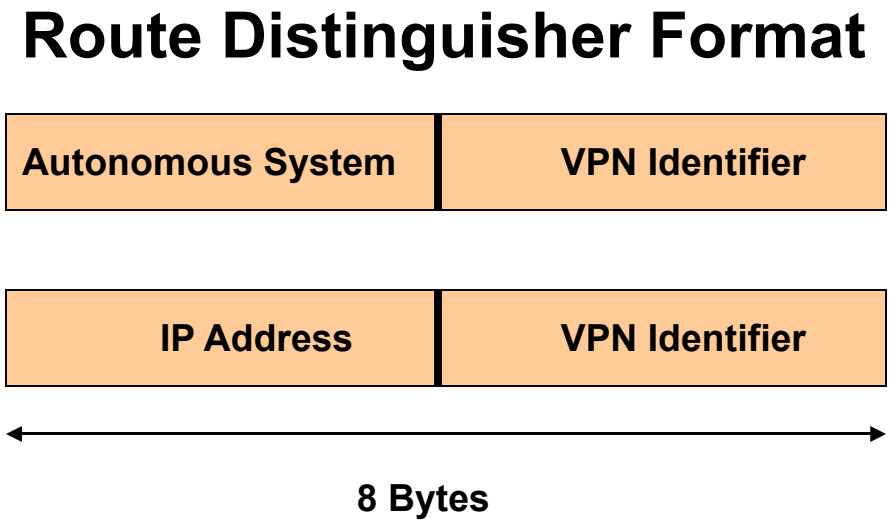
The 64-bit route distinguisher (RD) is prepended to an IPv4 address to make it globally unique.

The resulting address is a VPNv4 address.

VPNv4 addresses are exchanged between PE routers via BGP.

BGP that supports address families other than IPv4 addresses is called Multiprotocol BGP (MP-BGP).

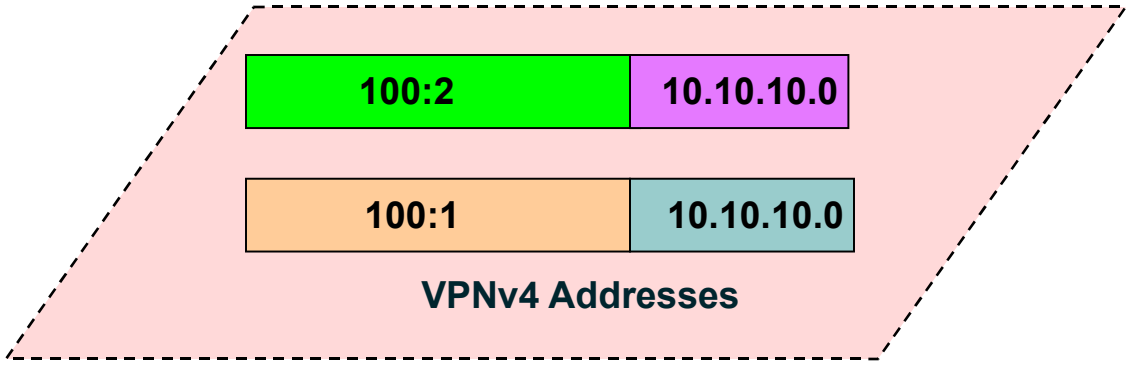
Route Distinguishers (Cont.)



Service Providers can use their BGP AS along with VPN customer identifier

Service Provider who do not have BGP AS, can use an IP address

Route Distinguishers (Cont.)



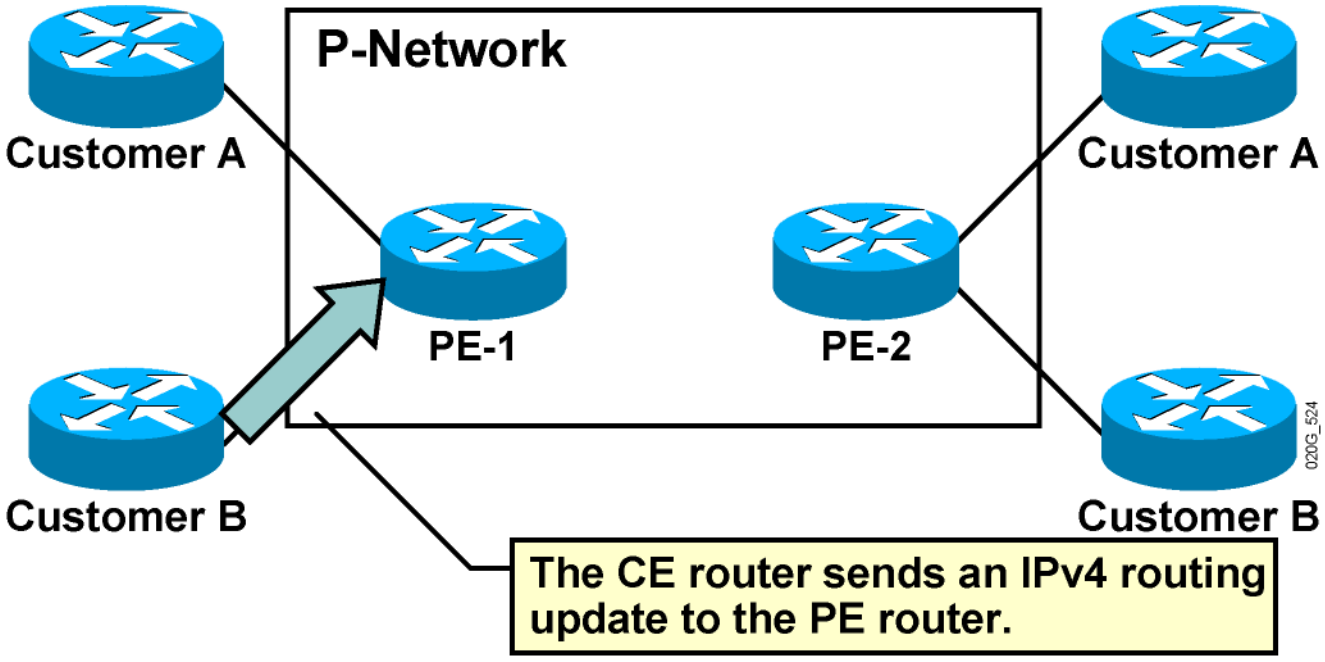
MP-iBGP update

Customer A has RD of 100:1

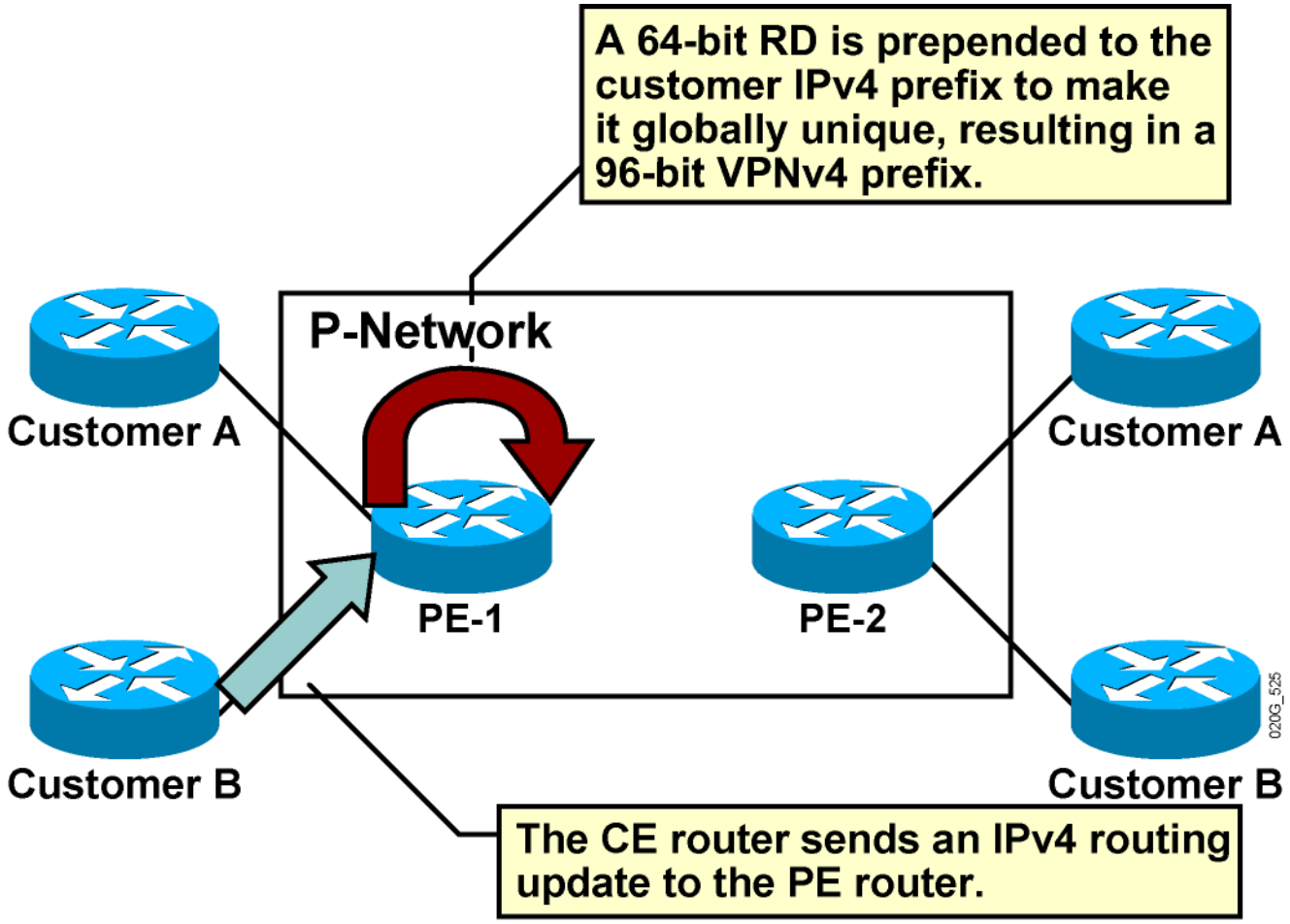
Customer B has RD of 100:2

Route Distinguisher keeps Customer A's update unique from Customer B in the MP-iBGP update, although they use the same IP address

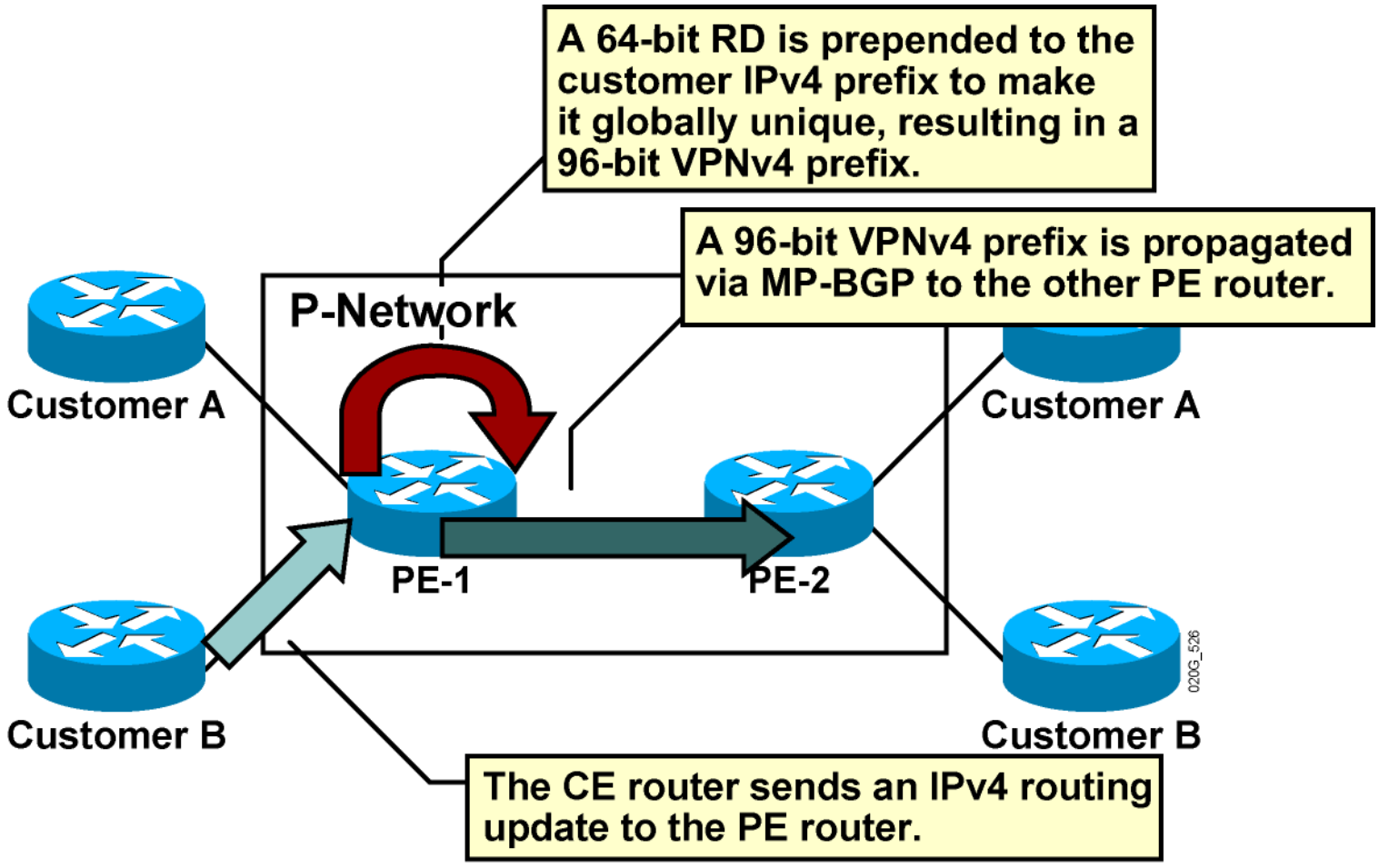
Route Distinguishers (Cont.)



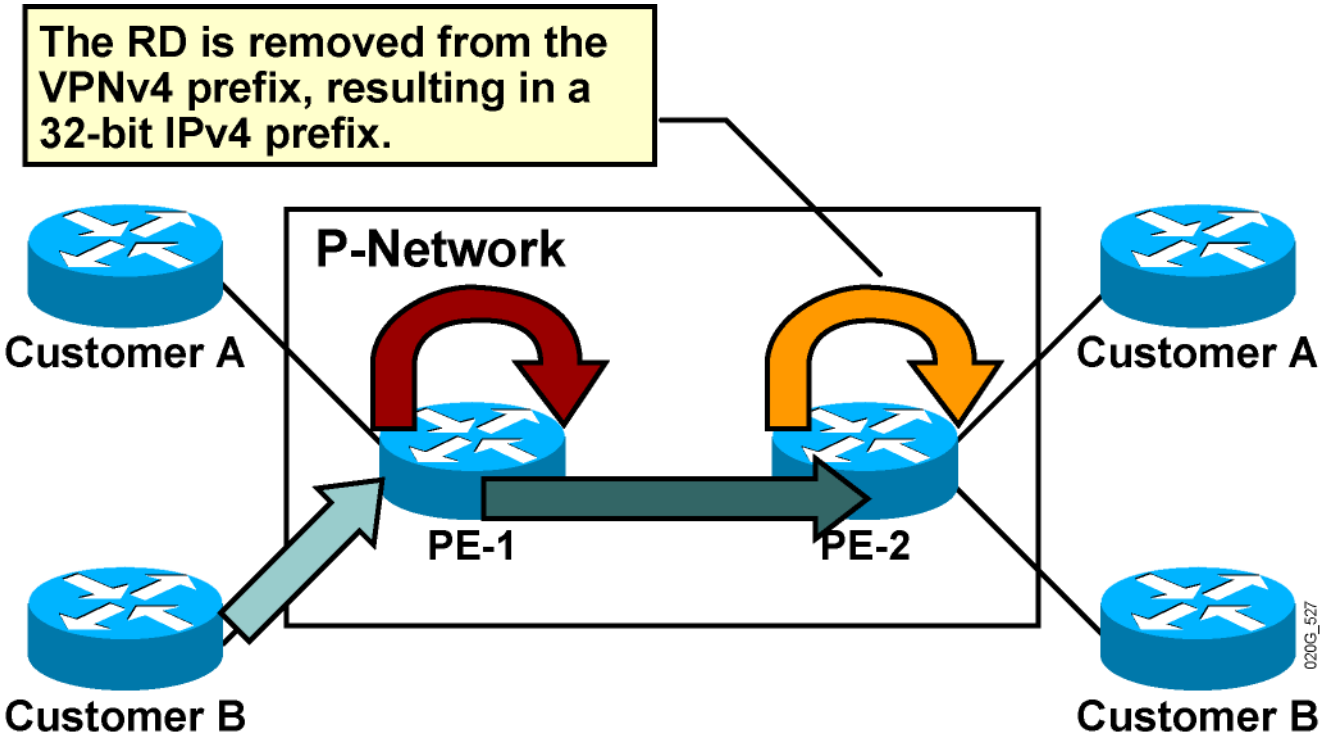
Route Distinguishers (Cont.)



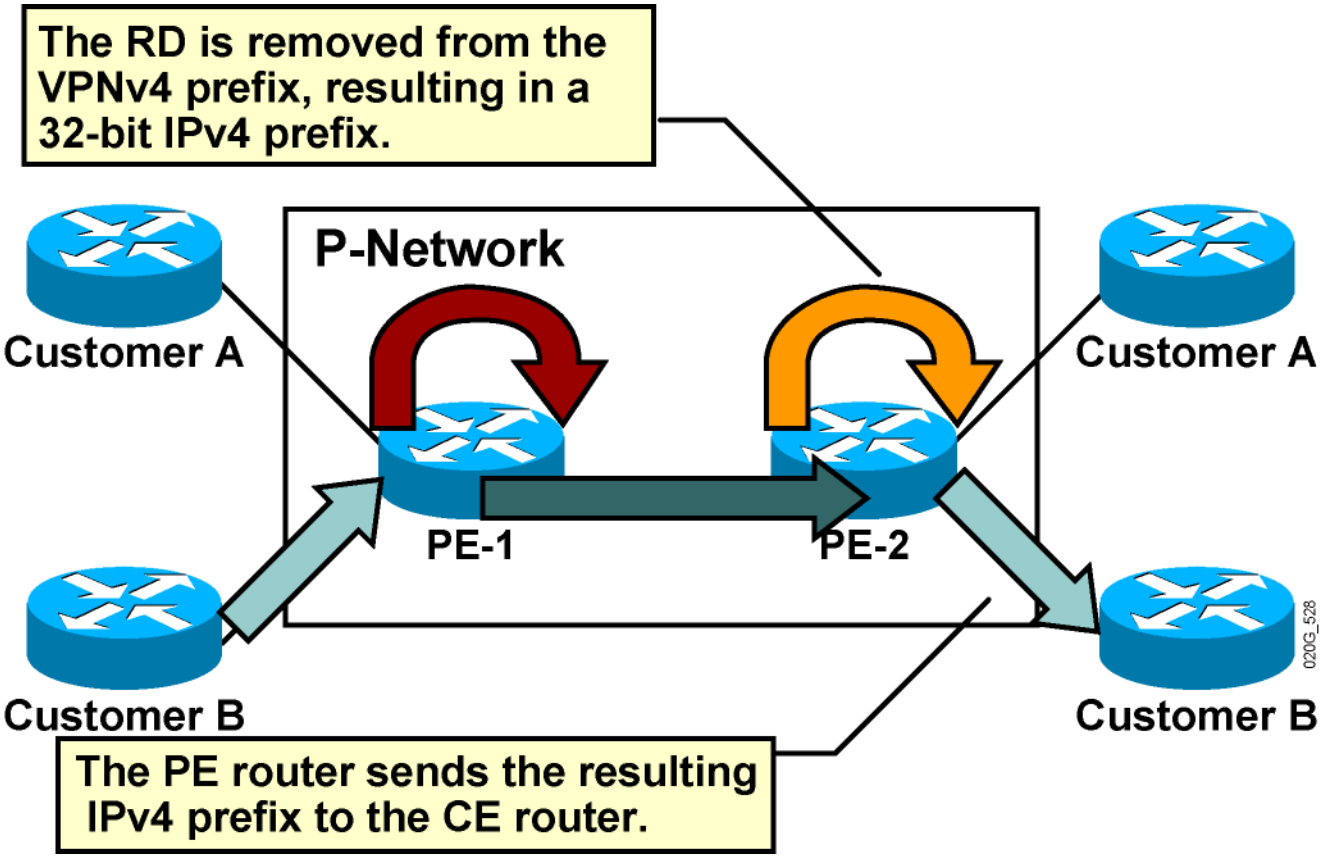
Route Distinguishers (Cont.)



Route Distinguishers (Cont.)



Route Distinguishers (Cont.)



Route Distinguishers (Cont.) Usage in an MPLS VPN

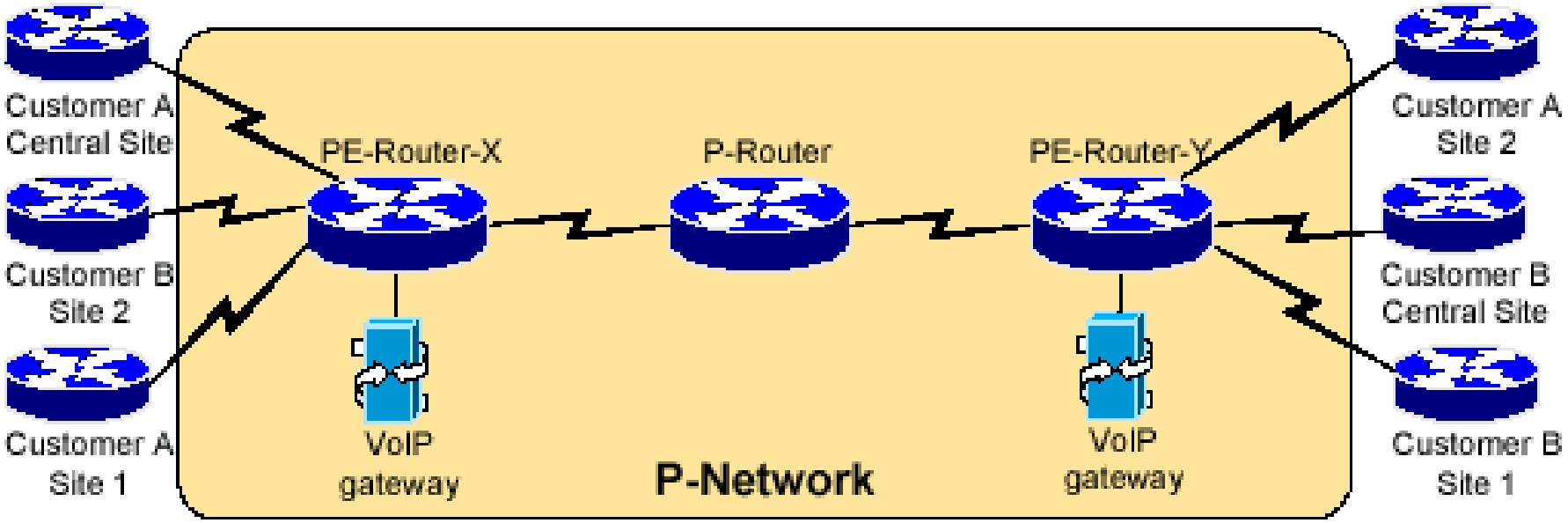
The RD has no special meaning.

Used only to make potentially overlapping IPv4 addresses globally unique.

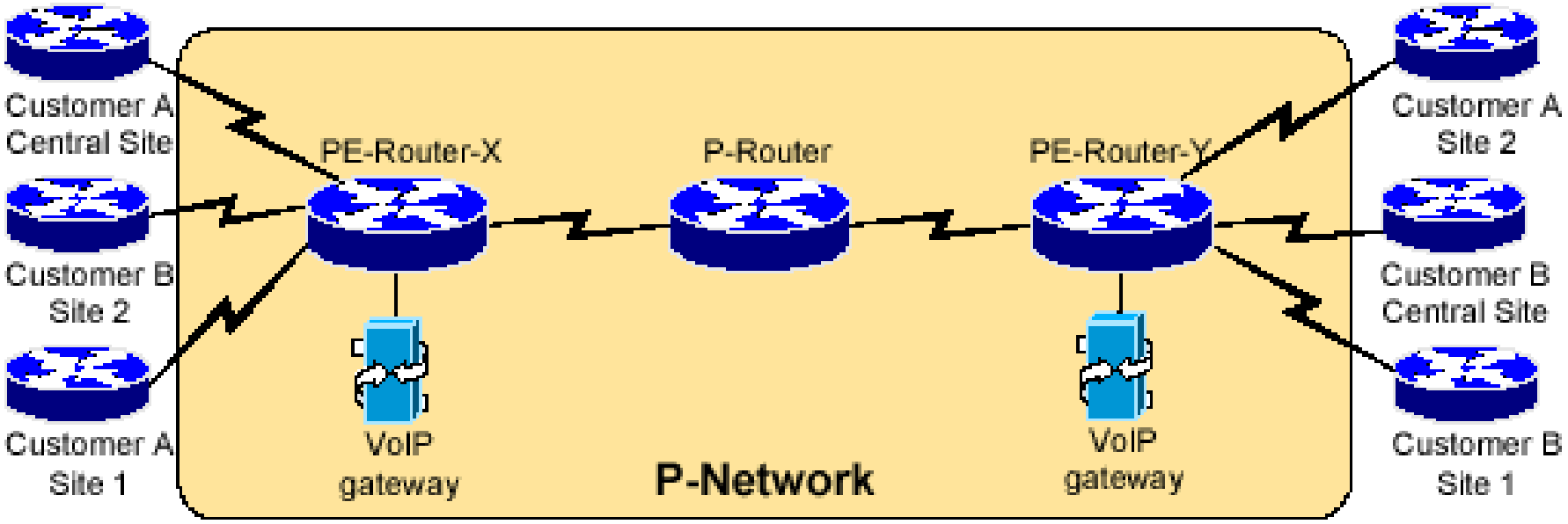
The RD could serve as a VPN identifier, but this design could not support all topologies required by the customers.

Route Targets
VoIP Service Sample

Why is RD not enough to identify VPNs?



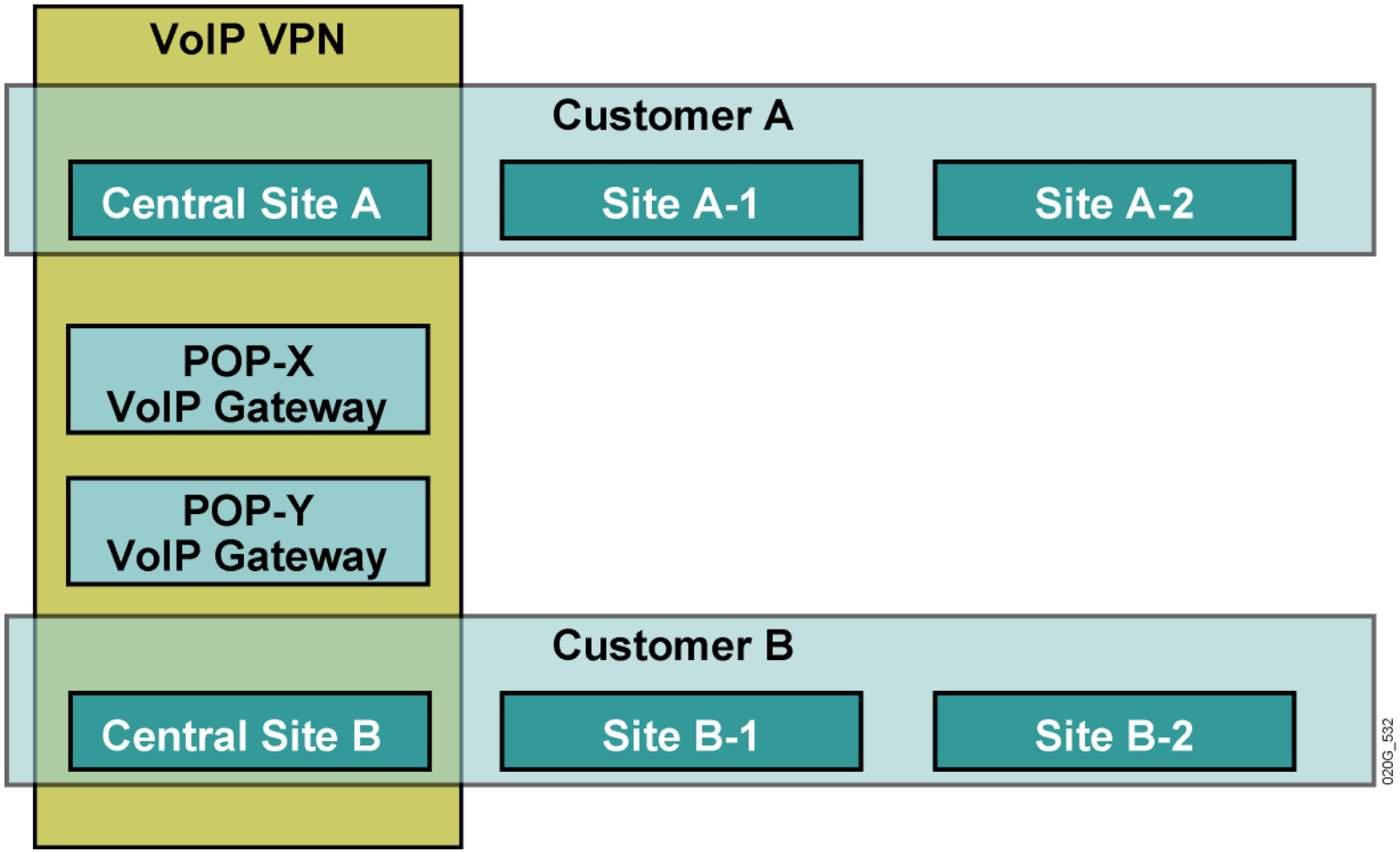
Route Targets
VoIP Service Sample



Requirements:

- All sites of one customer need to communicate.
- Central sites of both customers need to communicate with VoIP gateways and other central sites.
- Other sites from different customers do not communicate with each other.

Route Targets (Cont.)
Connectivity Requirements



Route Targets (Cont.)
Why Are They Needed?

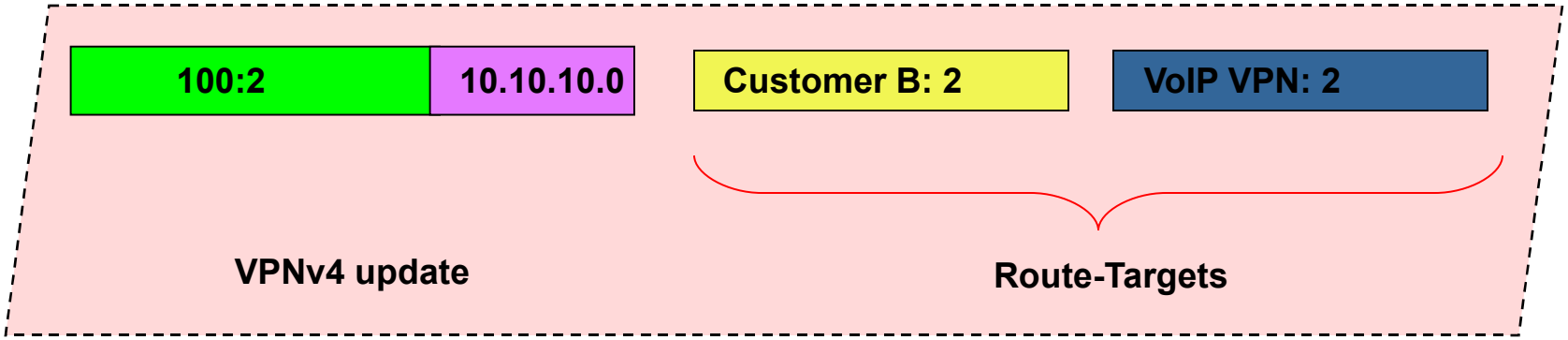
Some sites have to participate in more than one VPN.

The RD cannot identify participation in more than one VPN.

RTs were introduced in the MPLS VPN architecture to support complex VPN topologies.

- A different method is needed in which a set of identifiers can be attached to a route.

Route Targets (Cont.) What Are They?



MP-iBGP update

RTs are additional attributes attached to VPNv4 BGP routes to indicate VPN membership.

Format is same as Route Distinguisher

Extended BGP communities are used to encode these attributes.

- Extended communities carry the meaning of the attribute together with its value.

Any number of RTs can be attached to a single route.

Route Targets (Cont.)
How Do They Work?

Export RTs:

- Identifying VPN membership
- Appended to the customer route when it is converted into a VPNv4 route

Import RTs:

- Associated with each virtual routing table
- Select routes to be inserted into the virtual routing table

Virtual Private Networks Redefined

- With the introduction of complex VPN topologies, VPNs have had to be redefined:
  - A VPN is a collection of sites sharing common routing information.
  - A site can be part of different VPNs.
  - A VPN can be seen as a community of interest (closed user group, or CUG).
  - Complex VPN topologies are supported by multiple virtual routing tables on the PE routers.

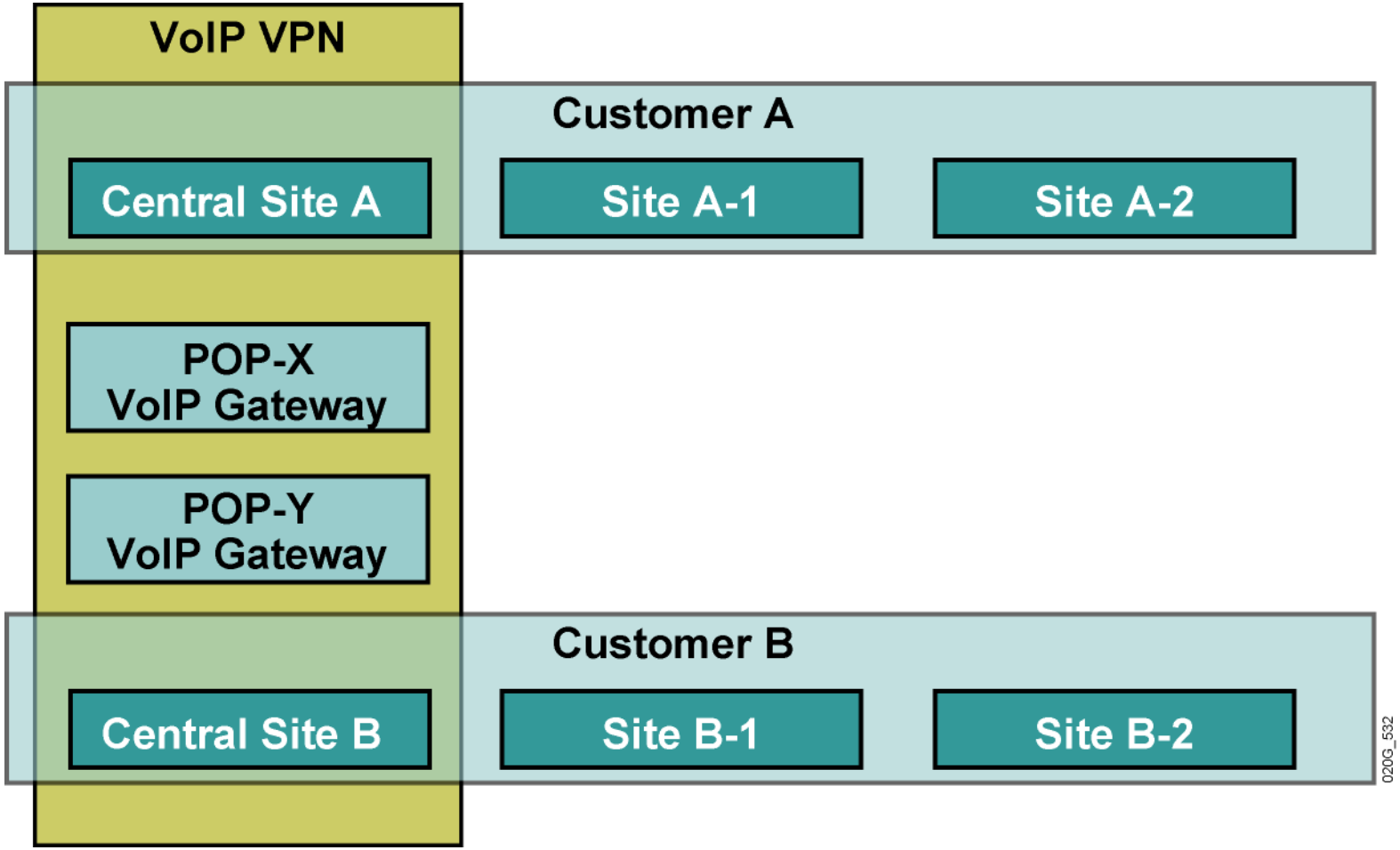
Impact of Complex VPN Topologies on Virtual Routing Tables

A virtual routing table in a PE router can be used only for sites with identical connectivity requirements.

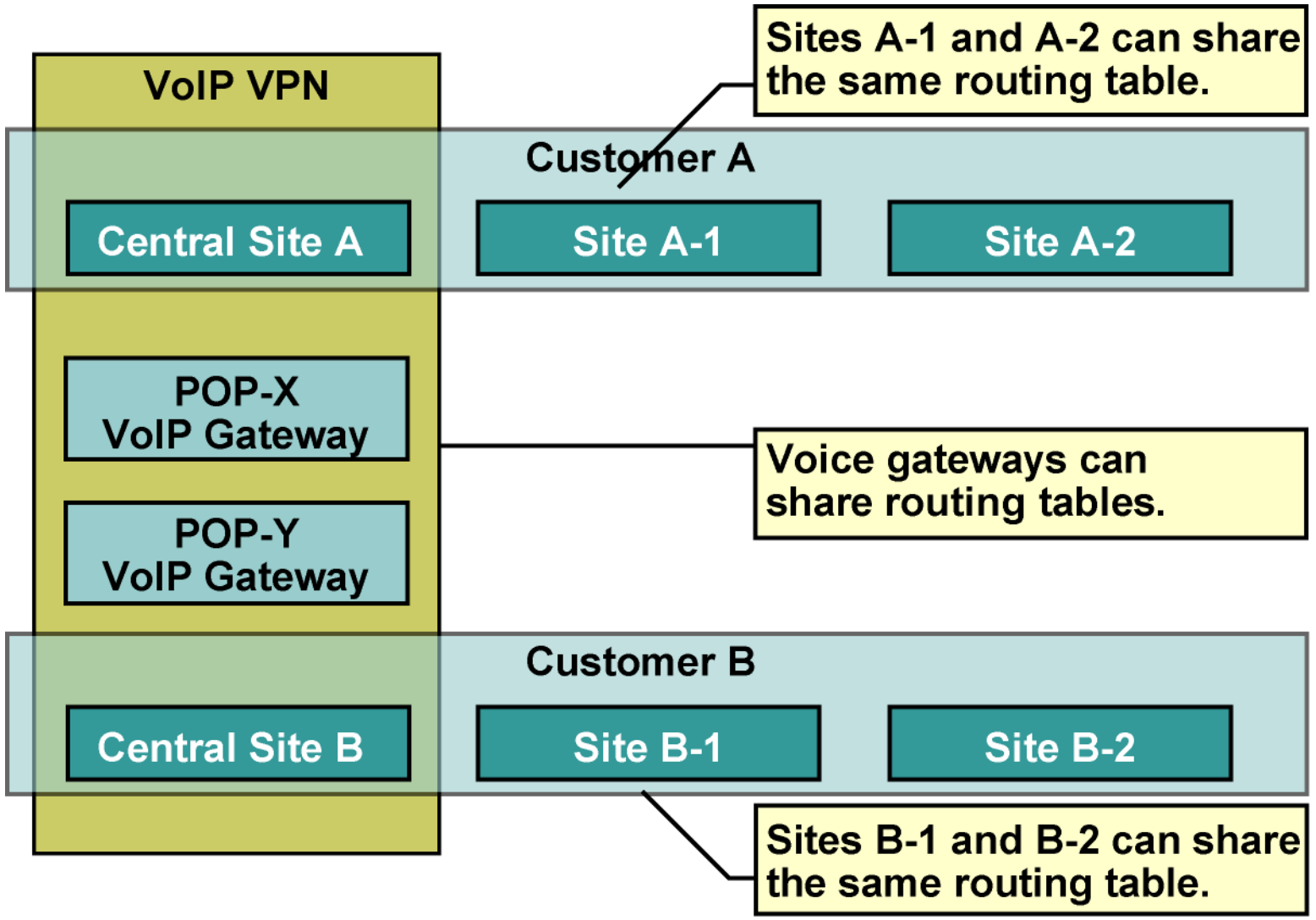
Complex VPN topologies require more than one virtual routing table per VPN.

As each virtual routing table requires a distinct RD value, the number of RDs in the MPLS VPN network increases.

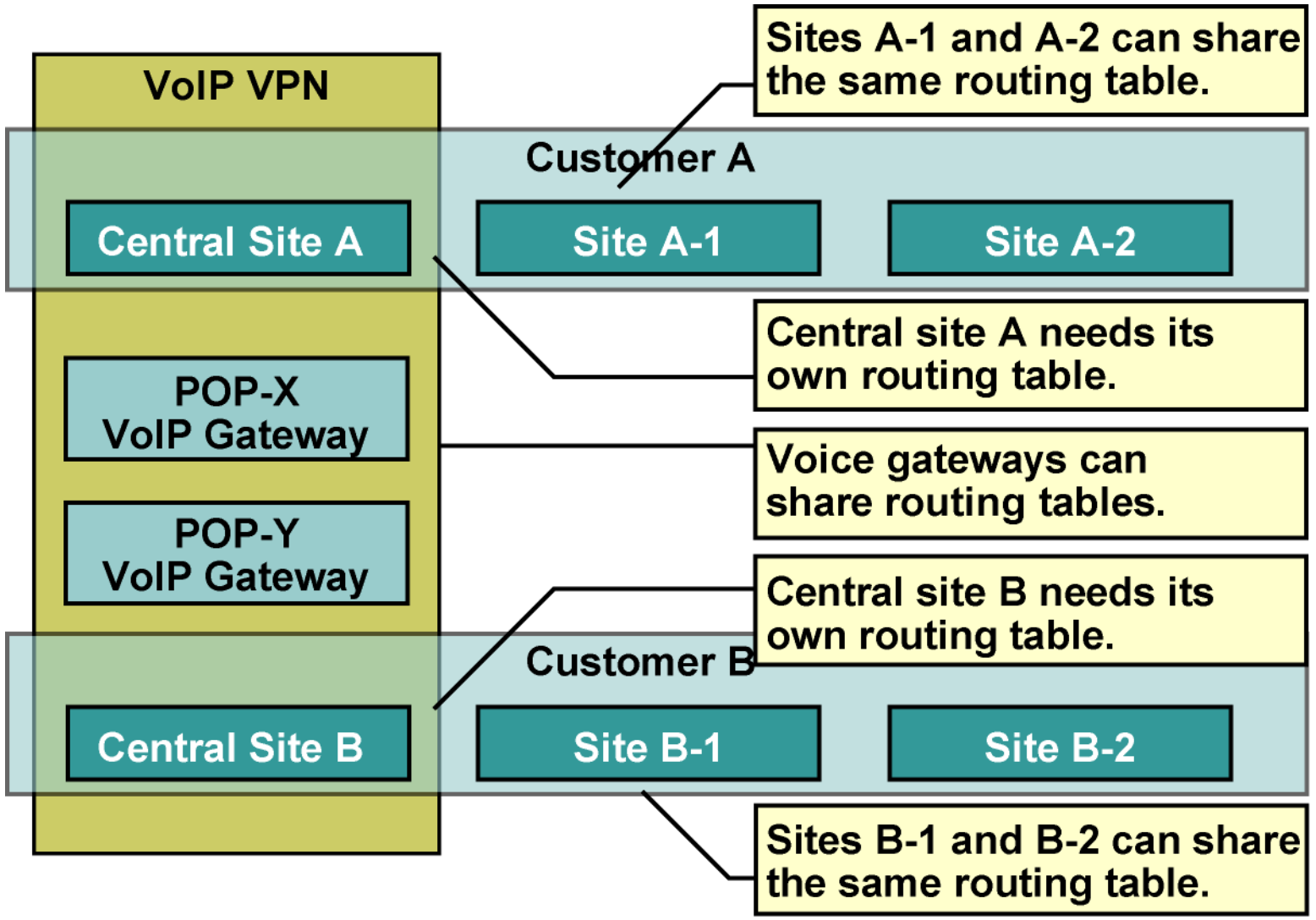
Impact of Complex VPN Topologies on Virtual Routing Tables (Cont.)



Impact of Complex VPN Topologies on Virtual Routing Tables (Cont.)



Impact of Complex VPN Topologies on Virtual Routing Tables (Cont.)



Important points to note for RT and RD

Route Distinguishers (RD) are only used to make ipv4 VPN addresses unique when advertising them over MP-iBGP, by making them vpnv4 prefixes

We can have one RD per vrf

Only one vrf can be assigned to an interface

Route Targets (RT) are used for VPN membership, so that complex scenarios can be addressed

VPN is the set of rules for customer connectivity and can be very complex

A VPN may have several RTs

Summary

MPLS VPN architecture combines the best features of the overlay and peer-to-peer VPN models.

Virtual routing tables are created for each customer.

BGP is used to exchange customer routes between PE routers.

Route distinguishers transform non-unique 32-bit addresses into 96-bit unique addresses.

Route targets are used to identify VPN membership in overlapping topologies.

Placing sites with different routing requirements in the same virtual routing table will result in inconsistent routing.

MPLS Bootcamp

MPLS VPN Routing Model

Outline

Overview
MPLS VPN Routing Requirements
MPLS VPN Routing
Support for Existing Internet Routing
Routing Tables on PE Routers
End-to-End Routing Update Flow
Route Distribution to CE Routers
Lesson Summary

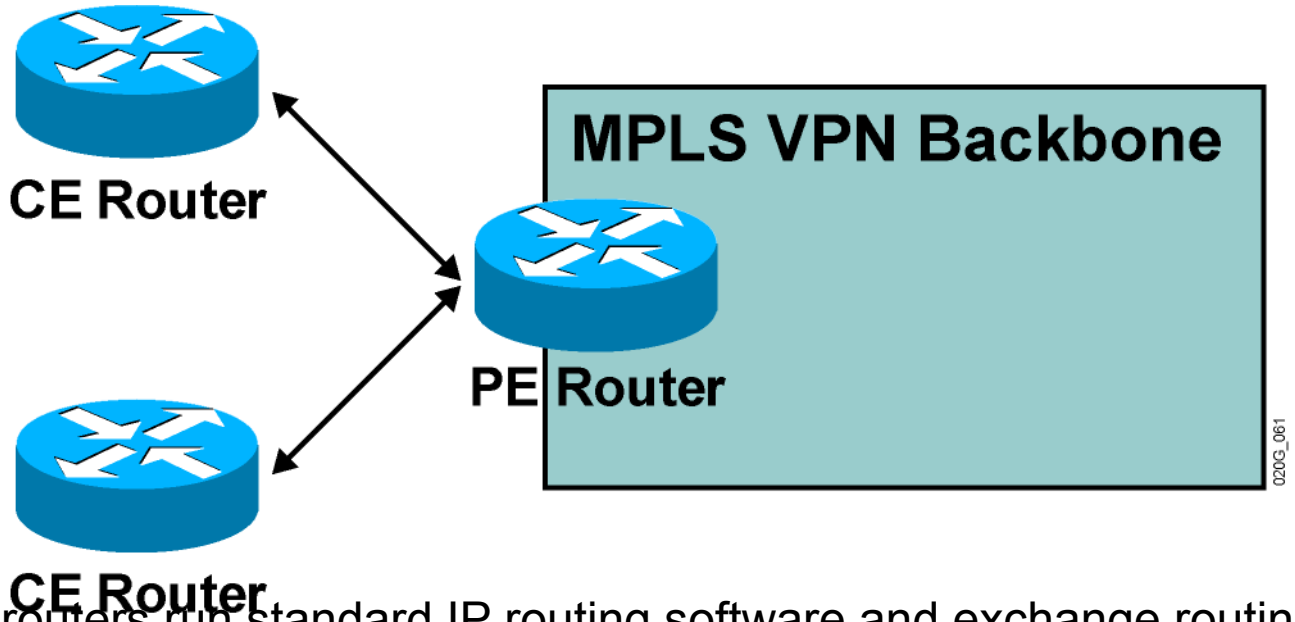
MPLS VPN Routing Requirements

CE routers have to run standard IP routing software.

PE routers have to support MPLS VPN services and Internet routing.

P routers have no VPN routes.

MPLS VPN Routing
CE Router Perspective

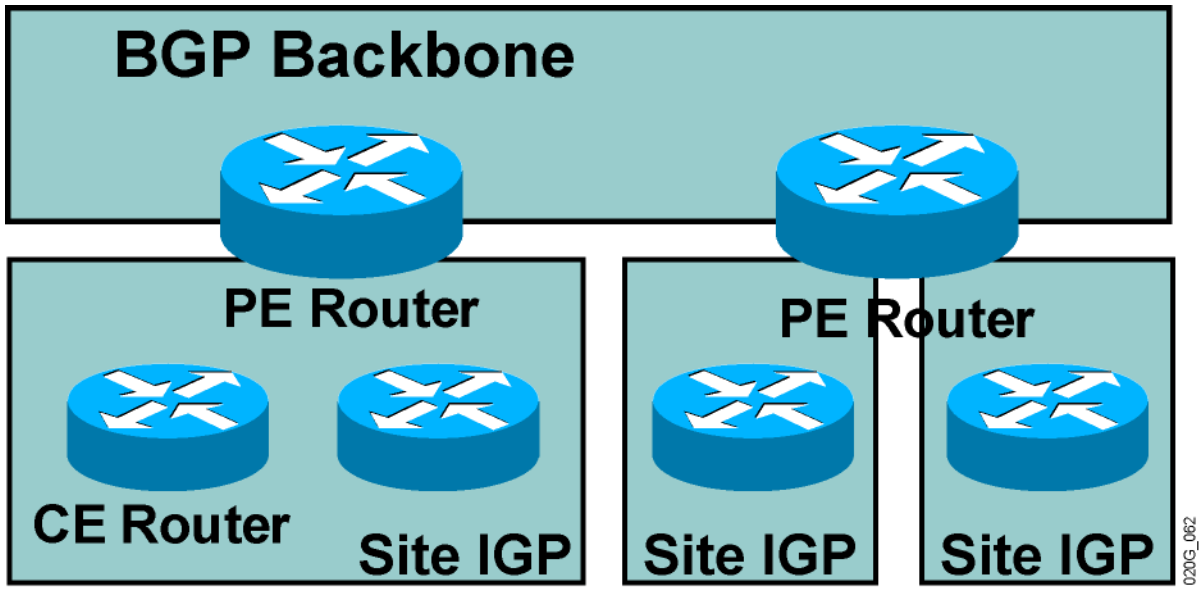


The CE routers run standard IP routing software and exchange routing updates with the PE router.

PE-CE protocols can be EBGP, OSPF, RIPv2, EIGRP, and static routes. ISIS support in the works

The PE router appears as another router in the C-network.

MPLS VPN Routing (cont.)
Overall Customer Perspective

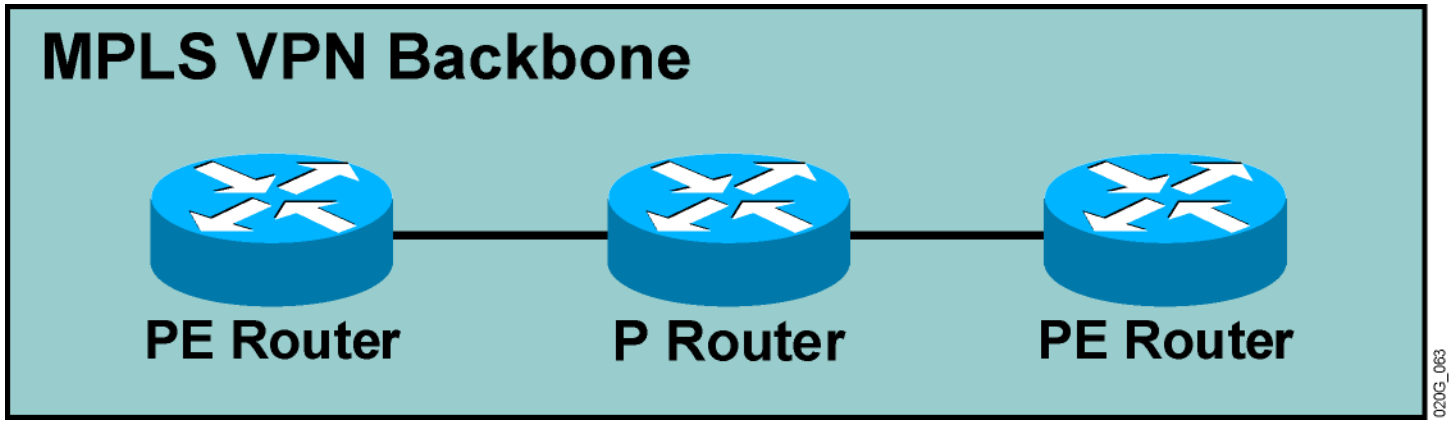


To the customer, the PE routers appear as core routers connected via a BGP backbone.

The usual BGP and IGP design rules apply.

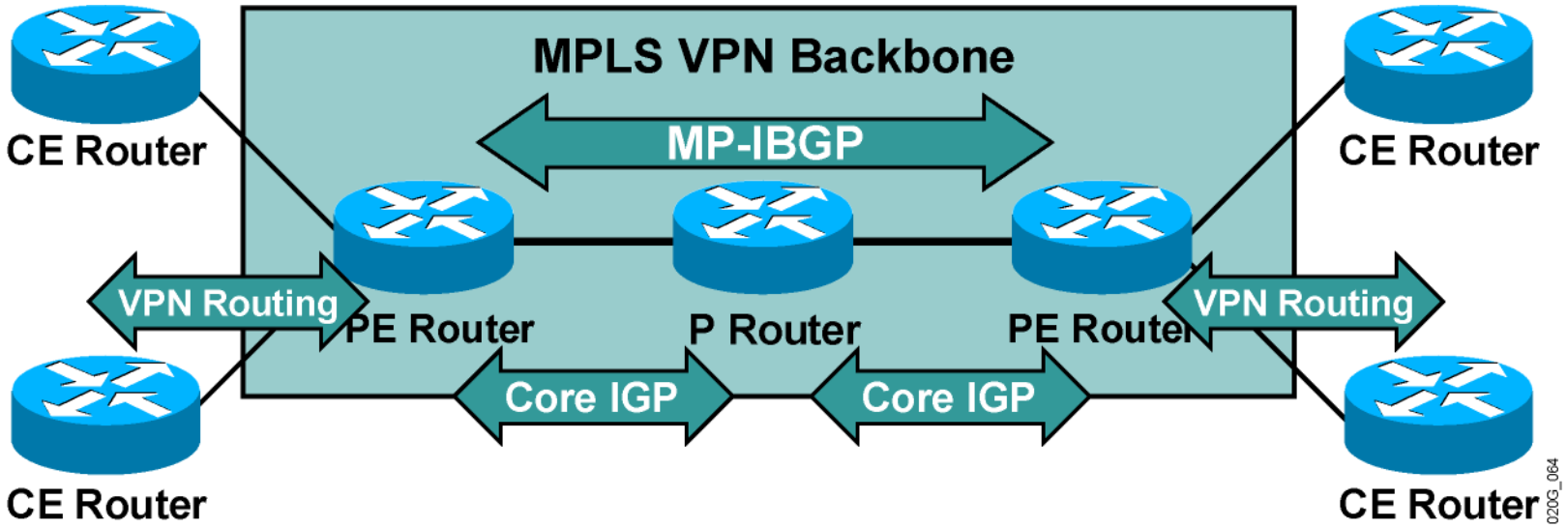
The P routers are hidden from the customer.

MPLS VPN Routing (Cont.)
P Router Perspective



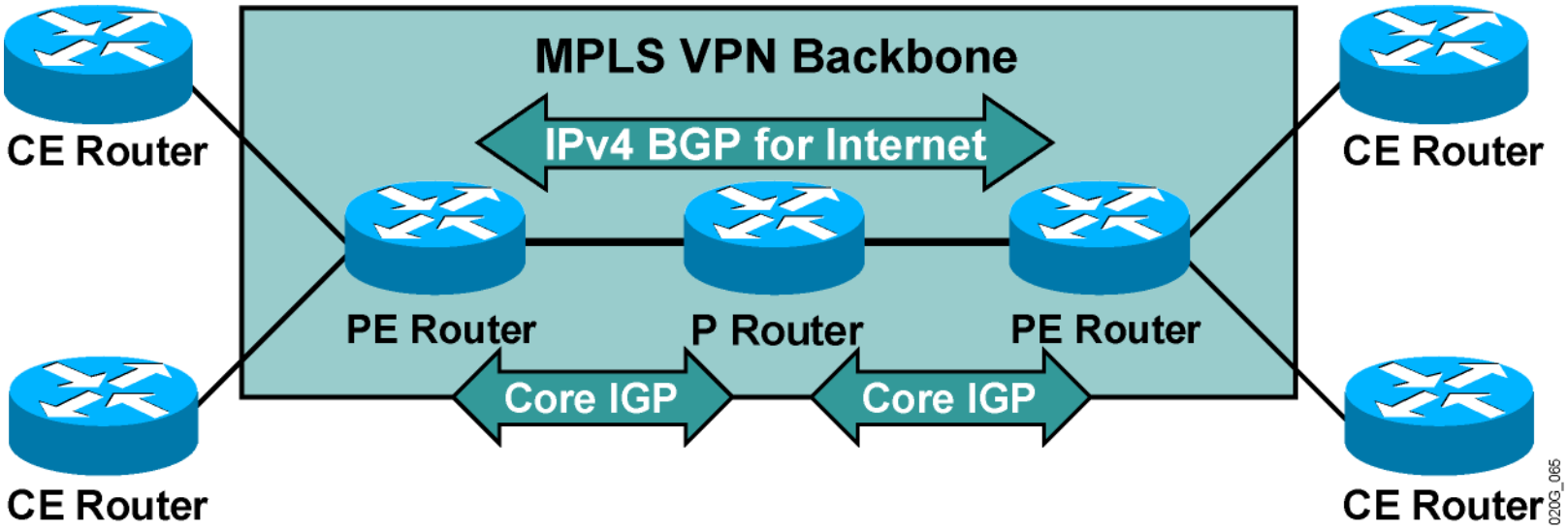
- P routers do not participate in MPLS VPN routing and do not carry VPN routes.
- P routers run backbone IGP with the PE routers and exchange information about global subnets (core links and loopbacks).

MPLS VPN Routing (Cont.)
PE Router Perspective



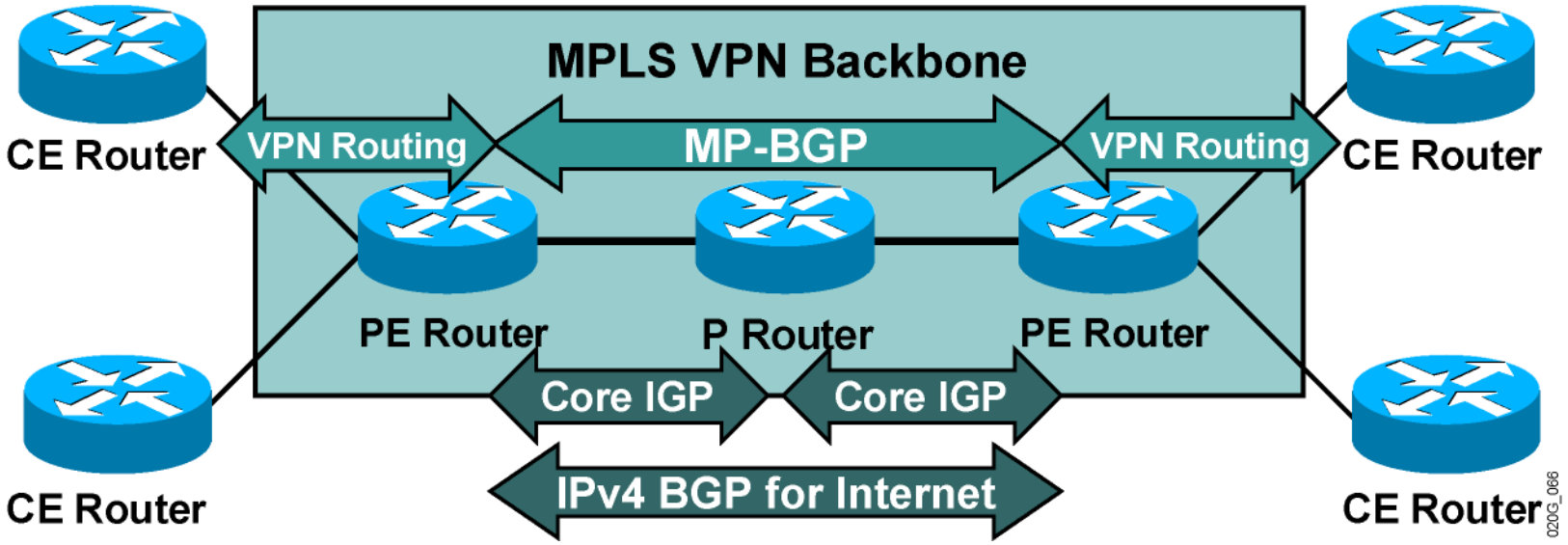
- PE routers:
  - Exchange VPN routes with CE routers via per-VPN routing protocols
  - Exchange core routes with P routers and PE routers via core IGP
  - Exchange VPNv4 routes with other PE routers via MP-IBGP sessions

Support for Existing Internet Routing



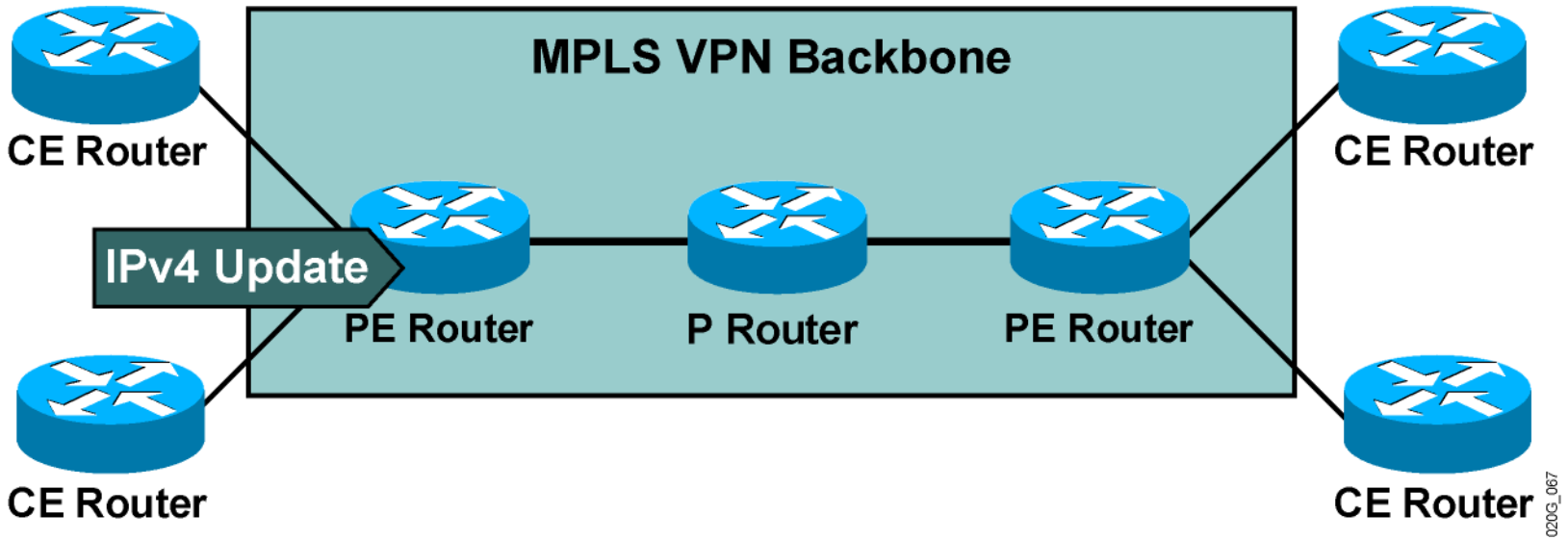
- PE routers can run standard IPv4 BGP in the global routing table:
  - PE routers exchange Internet routes with other PE routers.
  - CE routers do not participate in Internet routing.
  - P routers do not need to participate in Internet routing.

Routing Tables on PE Routers



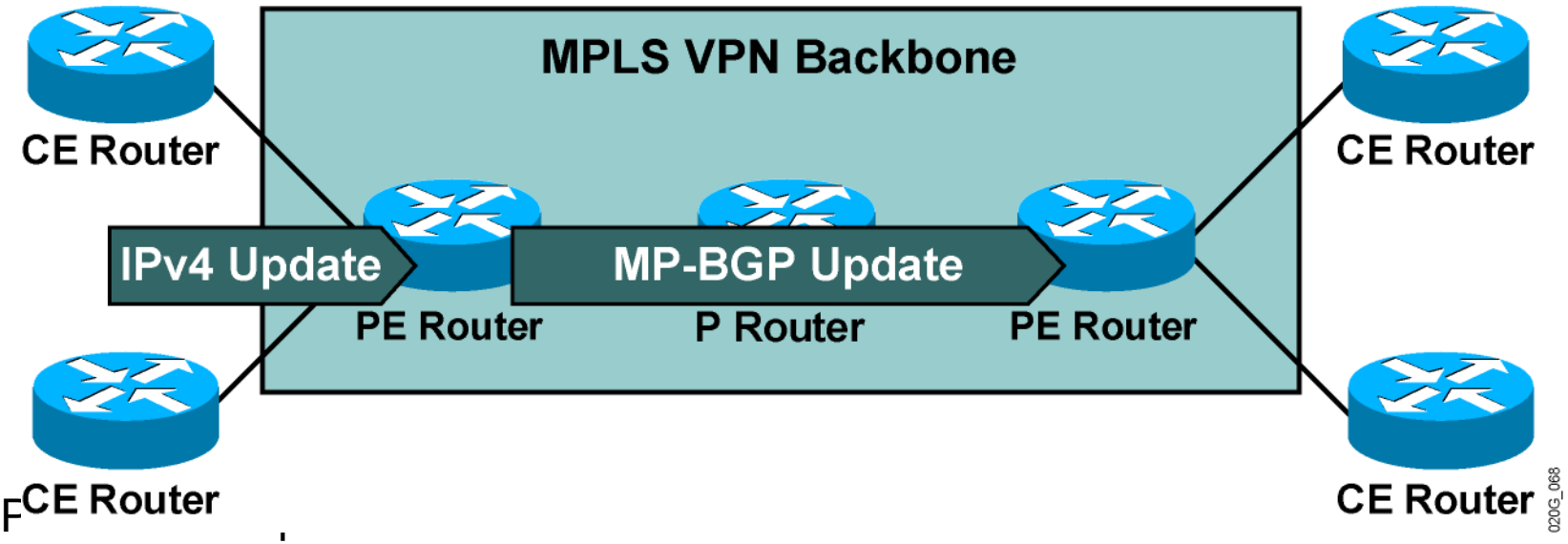
- PE routers contain a number of routing tables:
  - Global routing table, which contains core routes (filled with core IGP) and Internet routes (filled with IPv4 BGP)
  - VRF tables for sets of sites with identical routing requirements
  - VRFs filled with information from CE routers and MP-BGP information from other PE routers

End-to-End Routing Update Flow



- PE routers receive IPv4 routing updates from CE routers and install them in the appropriate VRF table.

End-to-End Routing Update Flow (Cont.)

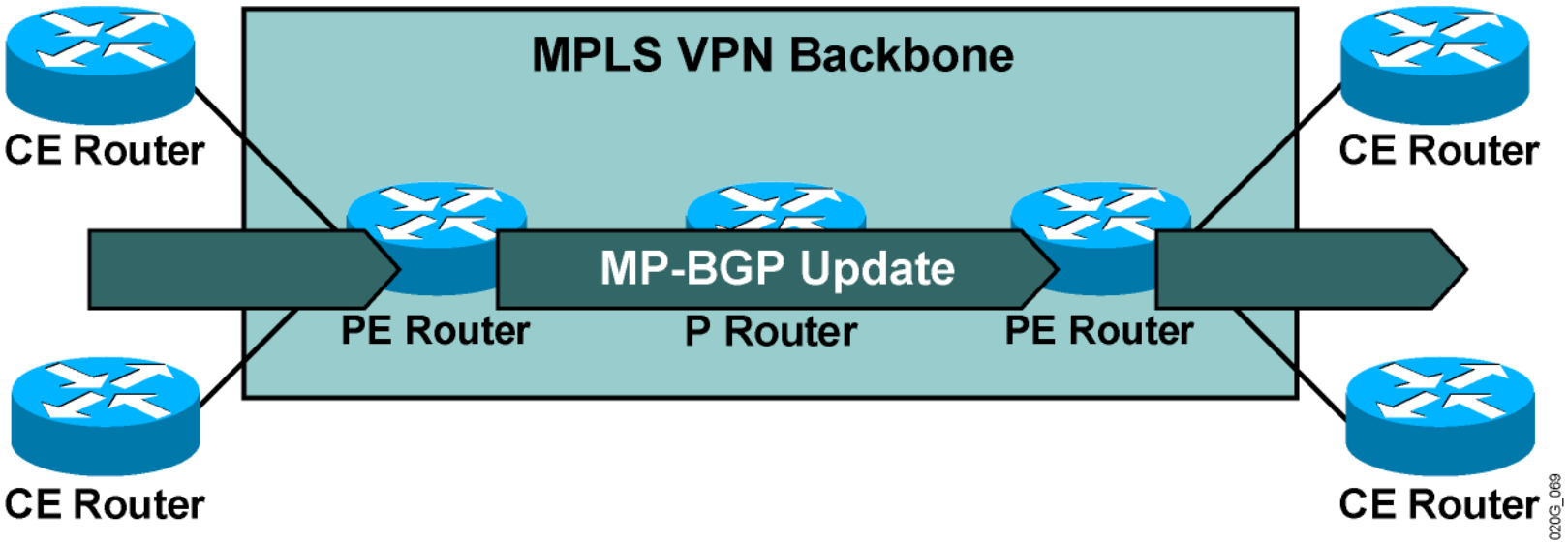


- FCE Router
  MP-BGP and propagate them as VPNv4 routes to other PE routers.

End-to-End Routing Update Flow (Cont.) MP-BGP Update

- An MP-BGP update contains the following:
  - VPNv4 address
  - Extended communities (route targets, optionally SOO)
  - Label used for VPN packet forwarding
  - Any other BGP attribute (for example, AS path, local preference, MED, standard community)

End-to-End Routing Update Flow (Cont.)



- Receiving PE router imports incoming VPNv4 routes into the appropriate VRF based on route targets attached to the routes.
- Routes installed in VRF are propagated to CE routers.

Route Distribution to CE Routers

Route distribution to sites is driven by the following:

- SOO
- RT BGP communities

A route is installed in the site VRF that matches the RT attribute.

Summary

MPLS VPNs technology does the following:

- Supports the use of standard IP routing between devices
- Provides scalable solutions
- Supports both MPLS VPNs and traditional Internet services

The internal service provider topology is transparent to the customer.

PE routers alone see all routing aspects of the MPLS VPN.

VRF tables contain sets of routes for sites with identical routing requirements.

Routes are transported using the following:

- IGP (internal core routes)
- BGP IPv4 (core Internet routes)
- BGP VPNv4 (PE-to-PE VPN routes)

MPLS Bootcamp

MPLS VPN Packet Forwarding

Outline

Overview

VPN Packet Forwarding Across an MPLS VPN Backbone

VPN Penultimate Hop Popping

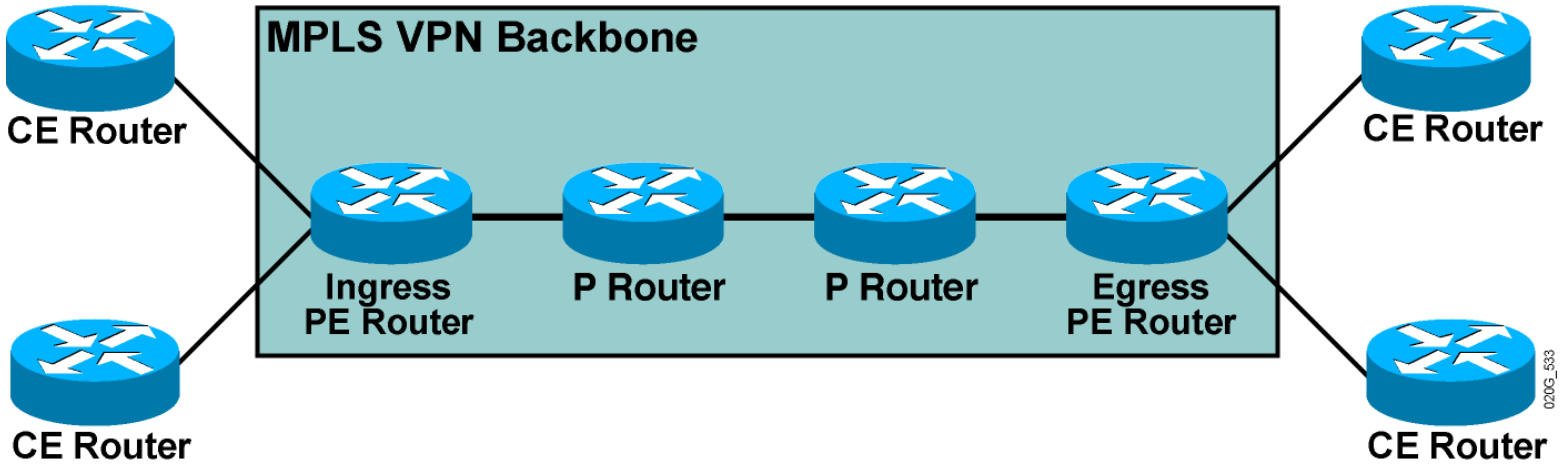
VPN Label Propagation

MPLS VPN and Label Propagation

MPLS VPN and Packet Forwarding

Lesson Summary

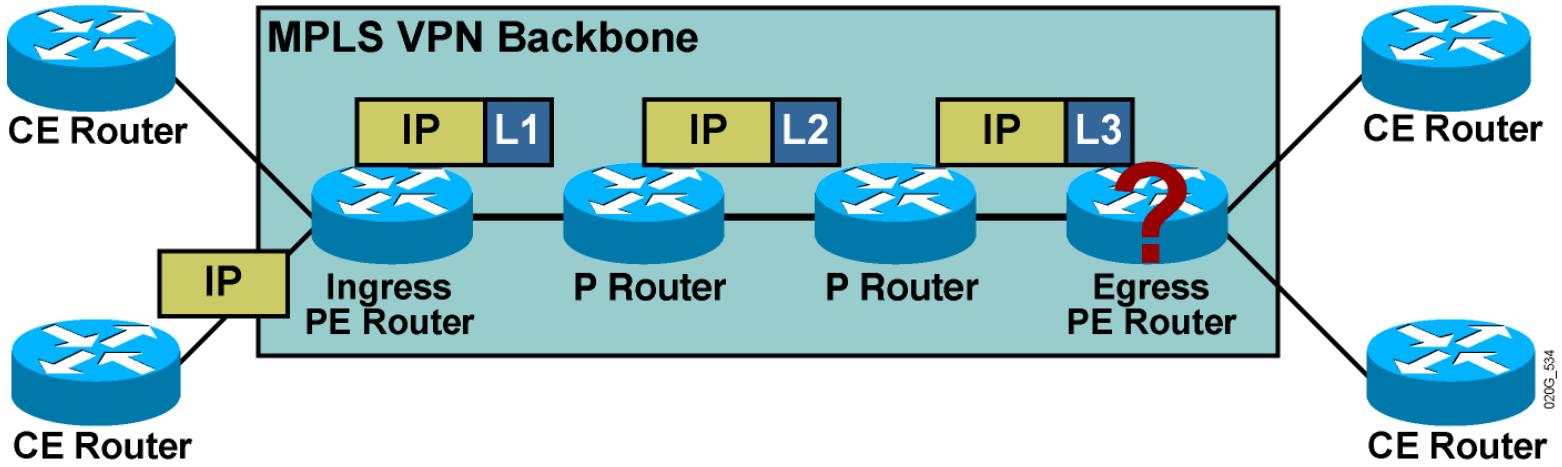
VPN Packet Forwarding Across an MPLS VPN Backbone



Question: How will the PE routers forward the VPN packets across the MPLS VPN backbone?

Answer #1: They will label the VPN packets with an LDP label for the egress PE router and forward the labeled packets across the MPLS backbone.

VPN Packet Forwarding Across an MPLS VPN Backbone



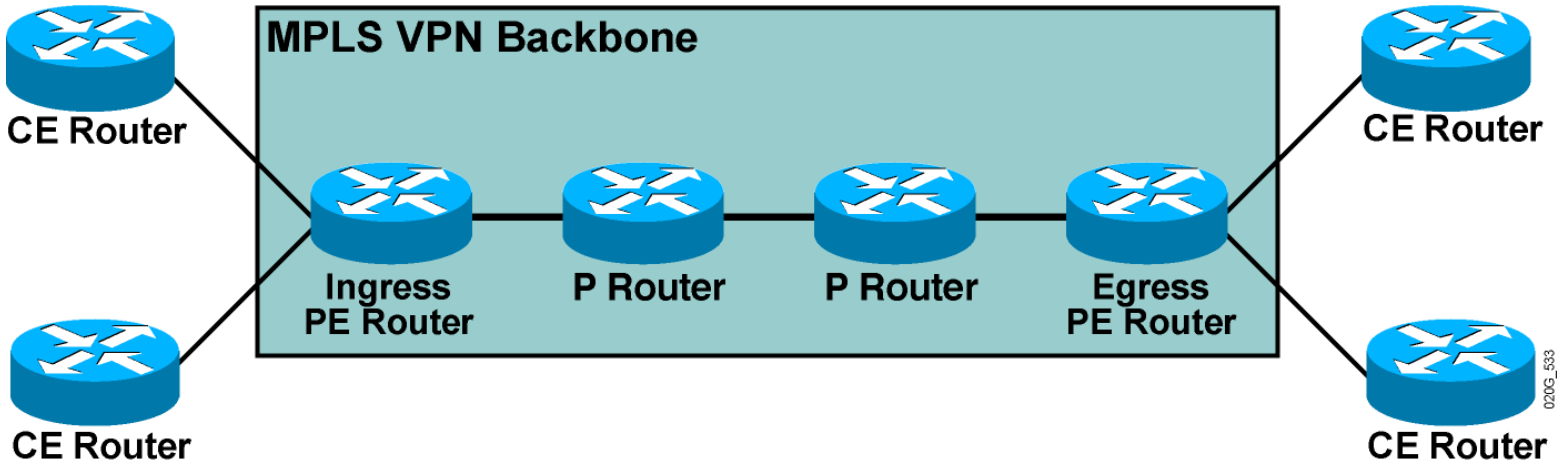
Question: How will the PE routers forward the VPN packets across the MPLS VPN backbone?

Answer #1: They will label the VPN packets with an LDP label for the egress PE router and forward the labeled packets across the MPLS backbone.

Results:

- The P routers perform the label switching, and the packet reaches the egress PE router.
- However, the egress PE router does not know which VRF to use for packet switching, so the packet is dropped.
- How about using a label stack?

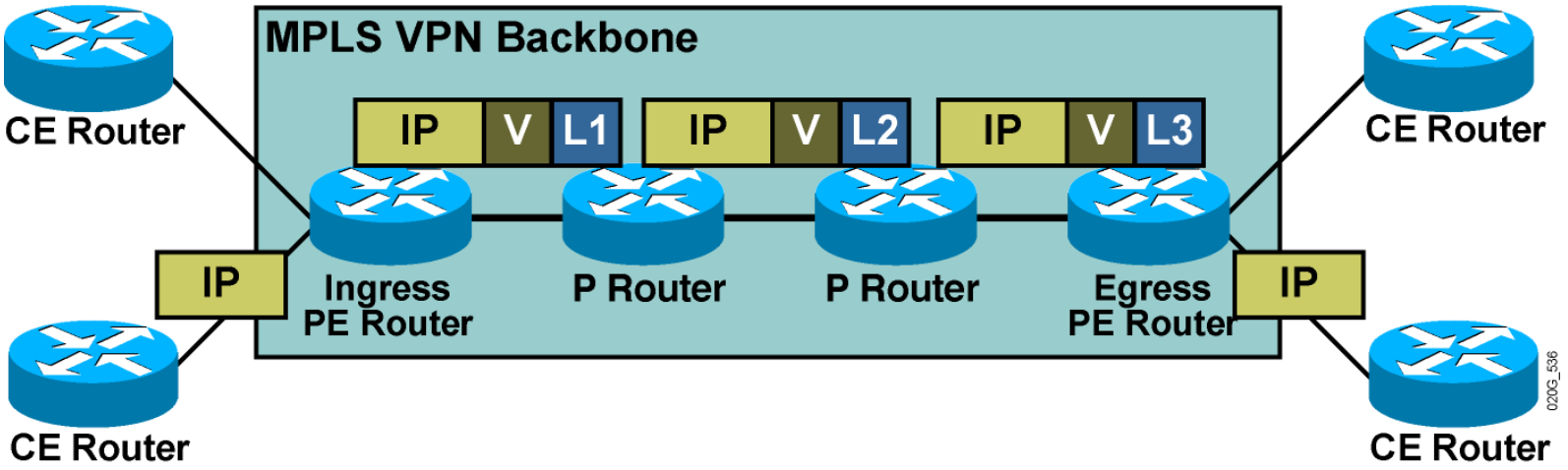
VPN Packet Forwarding Across an MPLS VPN Backbone (Cont.)



Question: How will the PE routers forward the VPN packets across the MPLS VPN backbone?

Answer #2: They will label the VPN packets with a label stack, using the LDP label for the egress PE router as the top label, and the VPN label assigned by the egress PE router as the second label in the stack.

VPN Packet Forwarding Across an MPLS VPN Backbone (Cont.)



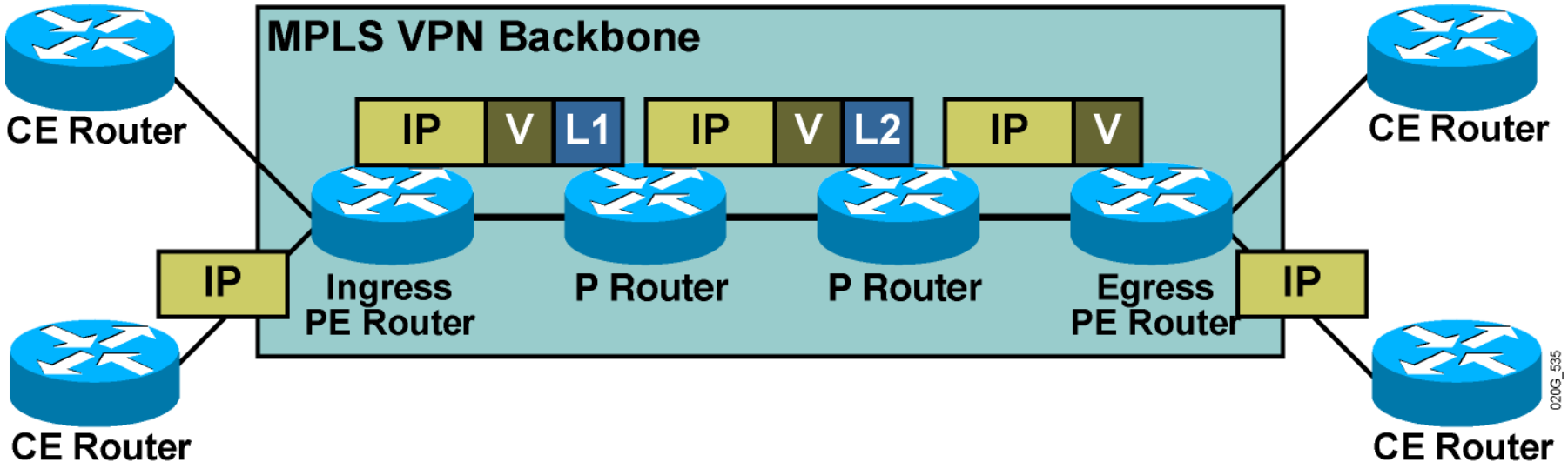
Question: How will the PE routers forward the VPN packets across the MPLS VPN backbone?

Answer #2: They will label the VPN packets with a label stack, using the LDP label for the egress PE router as the top label, and the VPN label assigned by the egress PE router as the second label in the stack.

Result:

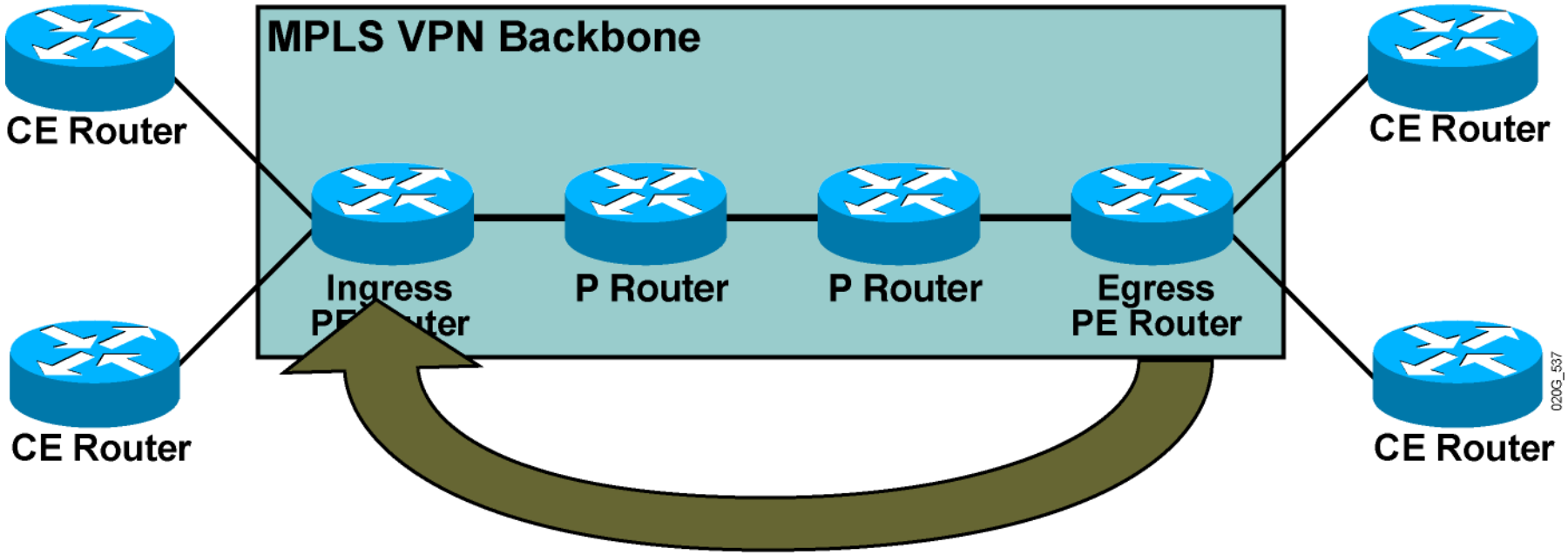
- The P routers perform label switching, and the packet reaches the egress PE router.
- The egress PE router performs a lookup on the VPN label and forwards the packet toward the CE router.

VPN Penultimate Hop Popping



- Penultimate hop popping on the LDP label can be performed on the last P router.
- The egress PE router performs label lookup only on the VPN label, resulting in faster and simpler label lookup.
- IP lookup is performed only once—in the ingress PE router.

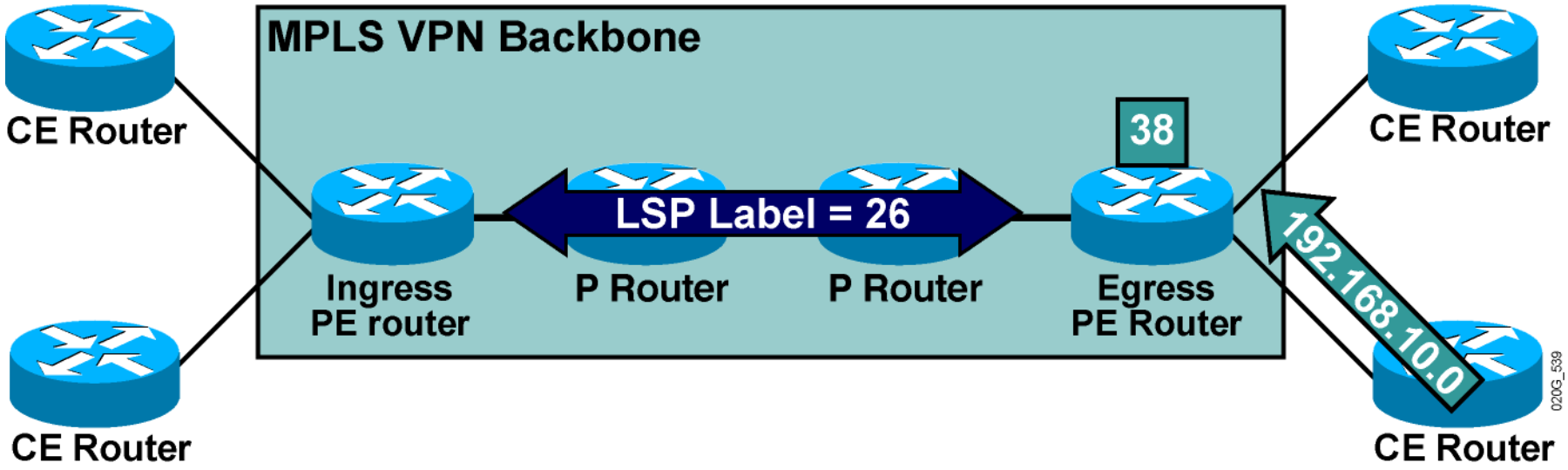
VPN Label Propagation



Question: How will the ingress PE router get the second label in the label stack from the egress PE router?

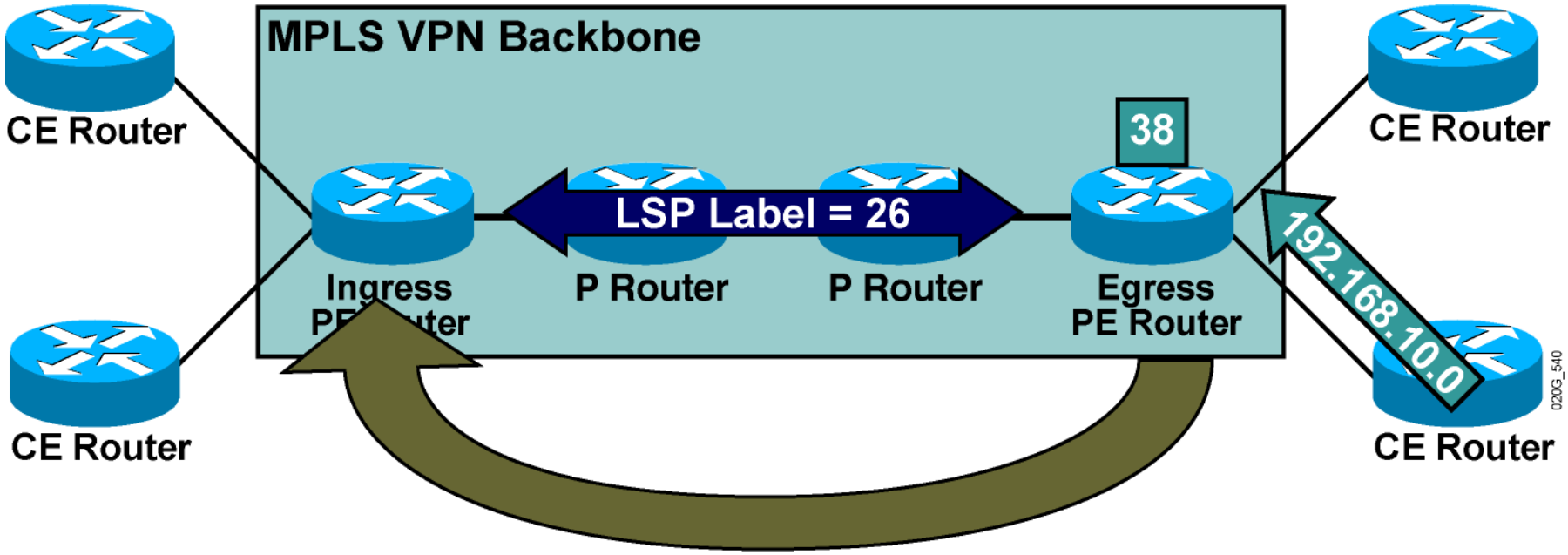
Answer: Labels are propagated in MP-BGP VPNv4 routing updates.

VPN Label Propagation (Cont.)



1: A VPN label is assigned to every VPN route by the egress PE router.

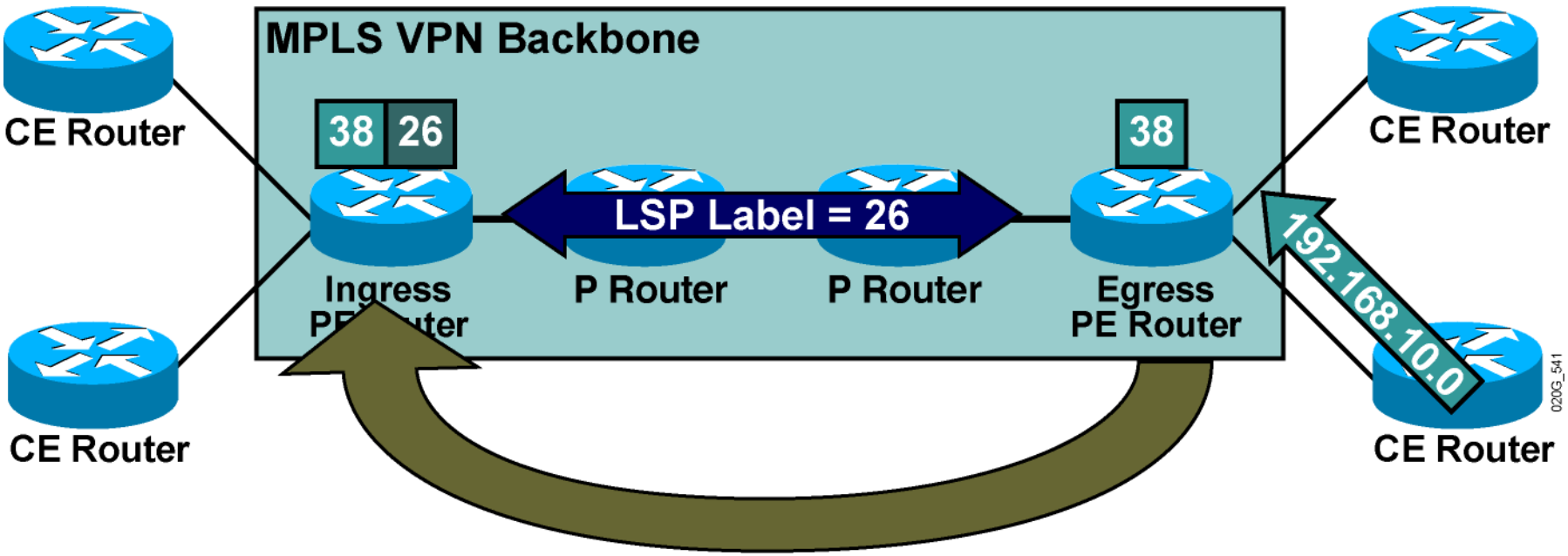
VPN Label Propagation (Cont.)



1: A VPN label is assigned to every VPN route by the egress PE router.

2: The VPN label is advertised to all other PE routers in an MP-BGP update.

VPN Label Propagation (Cont.)

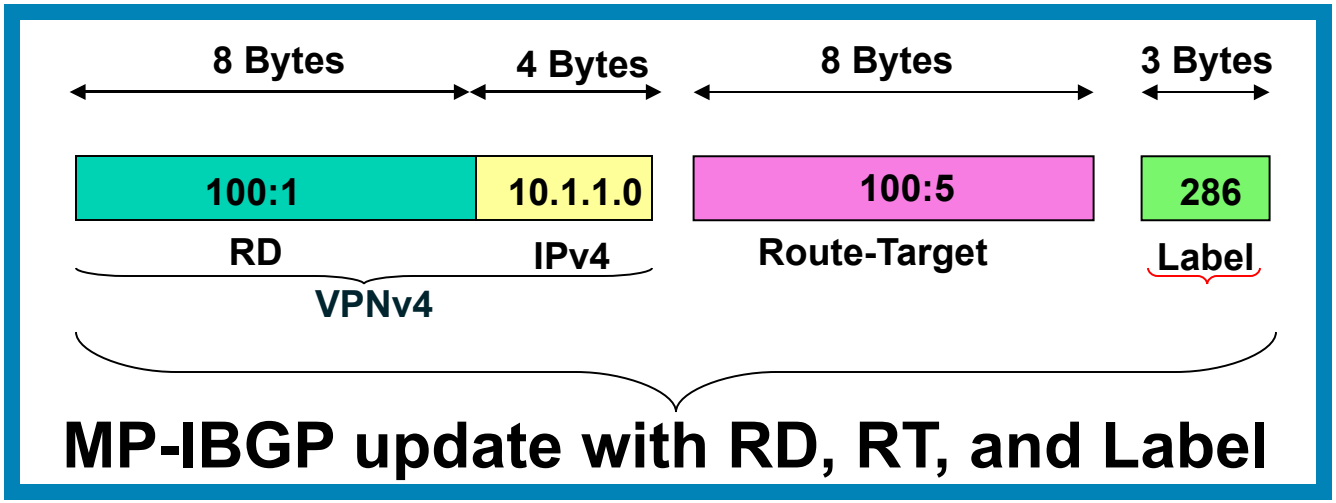


1: A VPN label is assigned to every VPN route by the egress PE router.

2: The VPN label is advertised to all other PE routers in an MP-BGP update.

3: A label stack is built in the VRF table.

VPN Label in MP-iBGP update



MPLS VPNs and Label Propagation

The VPN label must be assigned by the BGP next hop.

The BGP next hop should not be changed in the MP-IBGP update propagation.

- Do not use **next-hop-self** on confederation boundaries.

The PE router must be the BGP next hop.

- Use **next-hop-self** on the PE router (default on current IOS)

The label must be reoriginated if the next hop is changed.

- A new label is assigned every time that the MP-BGP update crosses the AS boundary where the next hop is changed.

MPLS VPNs and Packet Forwarding

The VPN label is understood only by the egress PE router.

An end-to-end LSP tunnel is required between the ingress and egress PE routers.

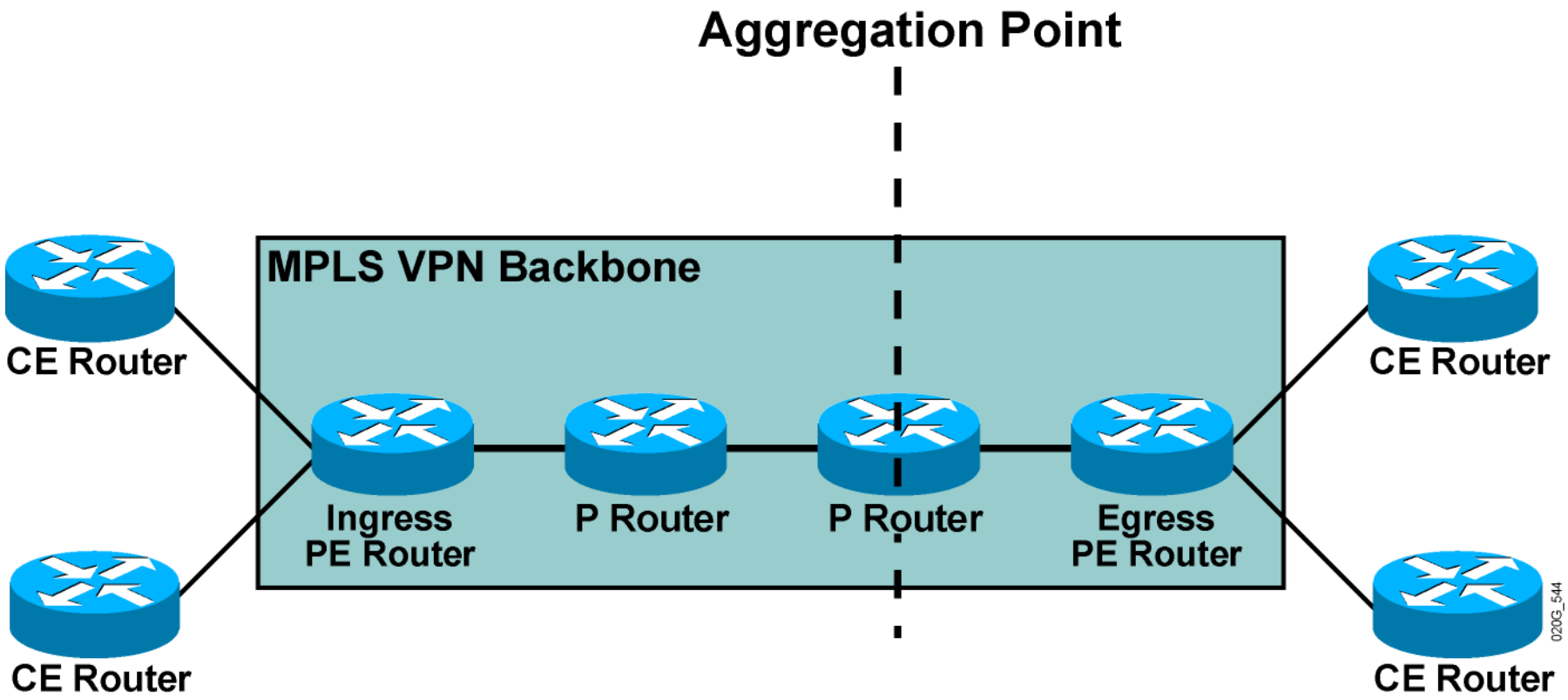
BGP next hops must not be announced as BGP routes.

LDP labels are not assigned to BGP routes.

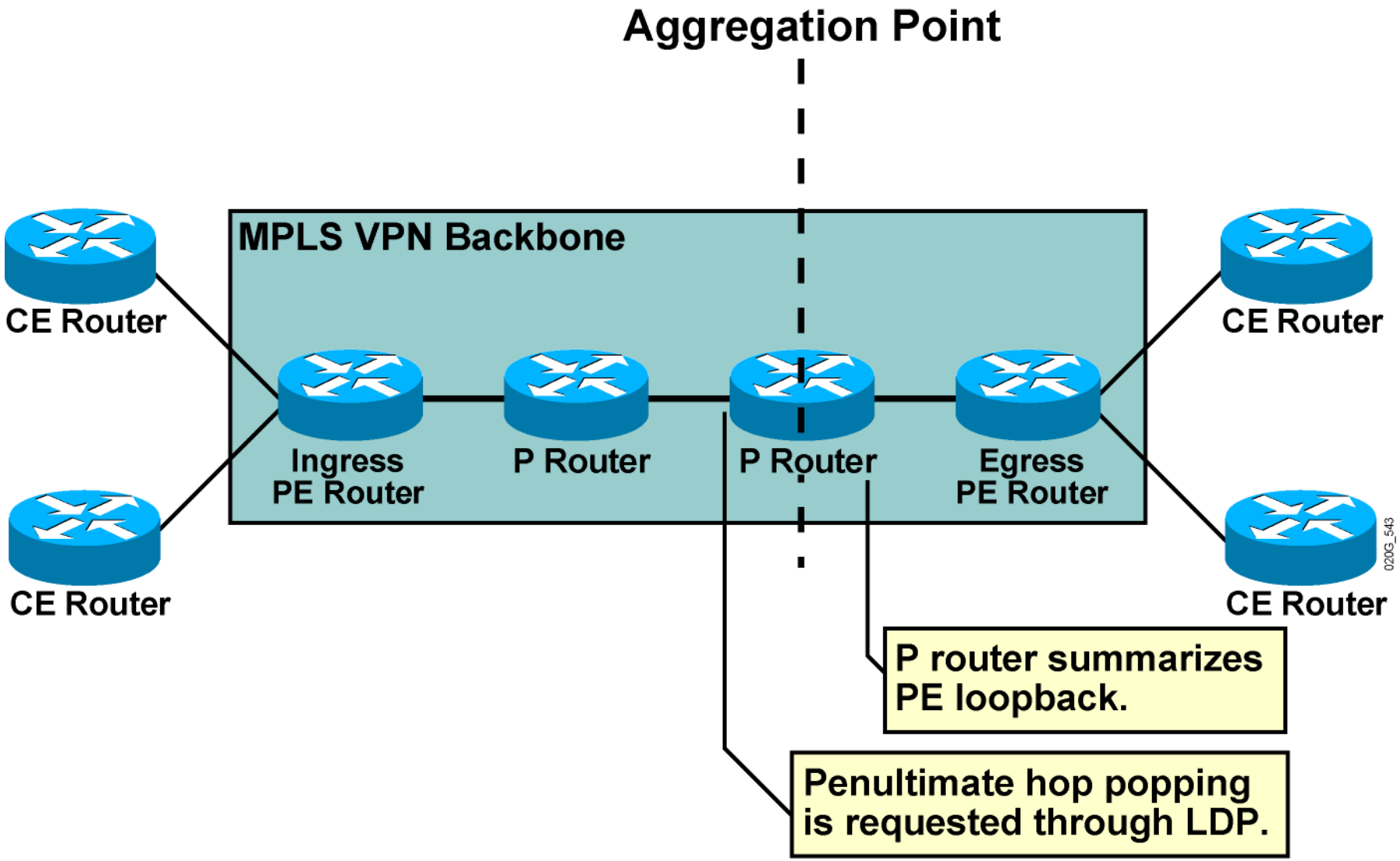
BGP next hops announced in IGP must not be summarized in the core network.

- Summarization breaks the LSP tunnel.

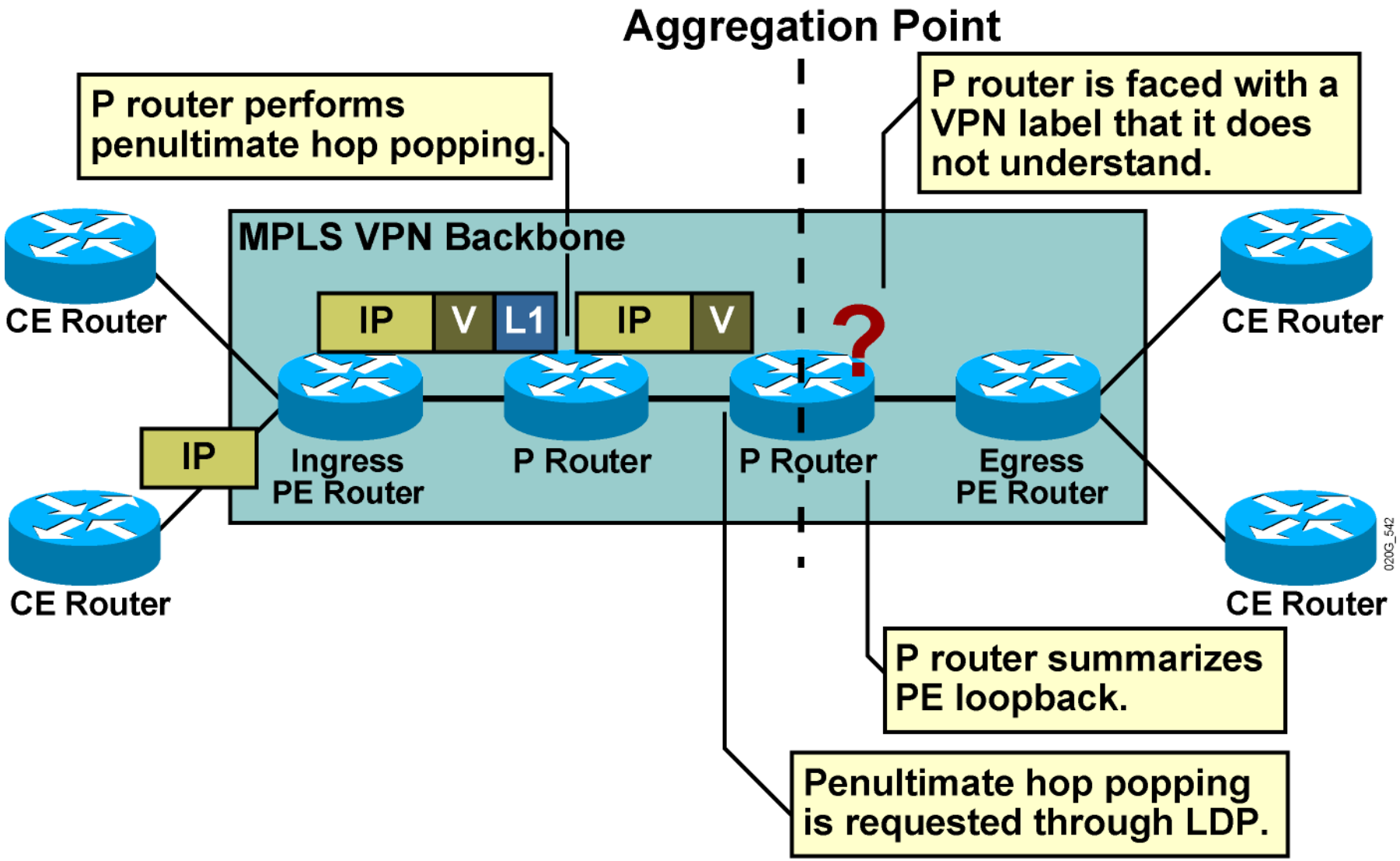
MPLS VPNs and Packet Forwarding (Cont.)
Summarization in the Core



MPLS VPNs and Packet Forwarding (Cont.)
Summarization in the Core



MPLS VPNs and Packet Forwarding (Cont.)
Summarization in the Core



Summary

PE routers forward packets across the MPLS VPN backbone using label stacking.

Labels are propagated between PE routers using MP-BGP.

BGP next hops should not be announced as BGP routes.

LDP labels are not assigned to BGP routes.