MPLS VPN

Peer to Peer VPN's

Agenda

- **MP-BGP**
- **VPN Overview**
- **MPLS VPN Architecture**
- **MPLS VPN Basic VPNs**
- **MPLS VPN Complex VPNs**
- **MPLS VPN Configuration (Cisco)**
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

 MPLS_VPN, v4.6 2

Multiprotocol BGP 1

- **BGP-4 (RFC 1771) is capable of carrying routing information only for IPv4**
- **The only three pieces of information carried by BGP-4 that are IPv4 specific are**
  - the NEXT_HOP attribute (expressed as an IPv4 address),
  - the AGGREGATOR (contains an IPv4 address)
  - the NLRI (expressed as IPv4 address prefixes)
- **Multiprotocol Extensions to BGP-4**
  - RFC 2858
  - enable it to carry routing information for multiple network layer protocols (e.g., IPv6, IPX, etc...).

 MPLS_VPN, v4.6 3

Multiprotocol BGP 2

- **To enable BGP-4 to support routing for multiple network layer protocols two things have to be added**
  - the ability to associate a particular network layer protocol with the next hop information
  - the ability to associate a particular network layer protocol with a NLRI
- **To identify individual network layer protocols**
  - Address Family Identifiers (AFI) are used
  - values defined in RFC 1700
    - RFC 1700 is historic, obsoleted by RFC 3232
    - RFC 3232 specifies a Online Database for ASSIGNED NUMBERS
      - www.iana.org

 MPLS_VPN, v4.6 4

Address Family Numbers (RFC 1700)

```
Number    Description
------    ------------------------------------------------------------
     0    Reserved
     1    IP (IP version 4)
     2    IP6 (IP version 6)
     3    NSAP
     4    HDLC (8-bit multidrop)
     5    BBN 1822
     6    802 (includes all 802 media plus Ethernet "canonical format")
     7    E.163
     8    E.164 (SMDS, Frame Relay, ATM)
     9    F.69 (Telex)
    10    X.121 (X.25, Frame Relay)
    11    IPX
    12    AppleTalk
    13    Decnet IV
    14    Banyan Vines
 65535    Reserved
```


Multiprotocol BGP 4

- **Address Family Identifier (AFI) in MP-BGP**
  - this parameter is used to differentiate routing updates of different protocols carried across the same BGP session
  - it is a 16-bit value
- **MP-BGP uses an additional Sub-Address Family Identifier (SAFI)**
  - it is a 8-bit value
    - 1 NLRI used for unicast forwarding
    - 2 NLRI used for multicast forwarding
    - 3 NLRI used for both unicast and multicast forwarding
- **Usual notation AFI/SAFI (i.e. x/y)**
    - 1/1 IP version 4 unicast
    - 1/2 IP version 4 multicast
    - 1/128 VPN-IPv4 unicast (used for MPLS-VPN)

Multiprotocol BGP 3

- **Capability Advertisement Procedures are used**
  - by a BGP speaker that to determine whether the speaker could use multiprotocol extensions with a particular peer or not -> RFC 3392
  - done during BGP Open with Capabilities Optional Parameter (Parameter Type 2)

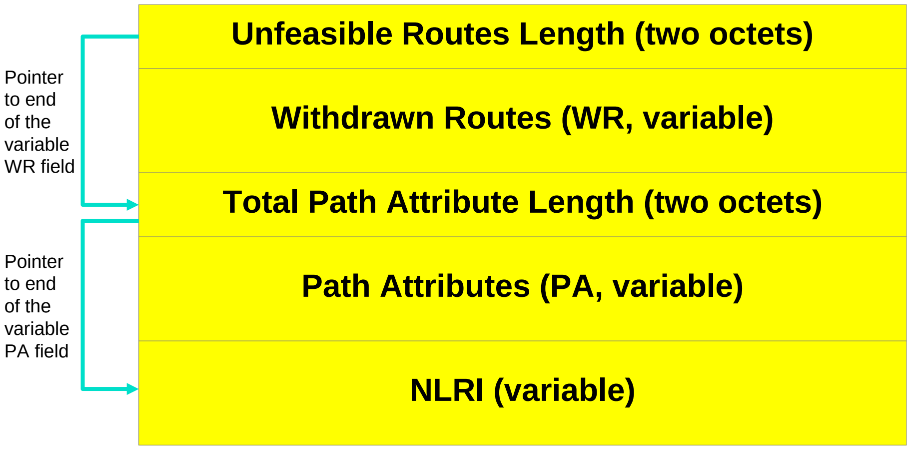
```
+------------------------------+
| Capability Code (1 octet)    |
+------------------------------+
| Capability Length (1 octet)  |
+------------------------------+
| Capability Value (variable)  |
+------------------------------+
```

  - Capability Code is unambiguously identifies individual capabilities. Capability Value is interpreted according to the value of the Capability Code field.

Multiprotocol BGP 4

- **Two new attributes**
  - Multiprotocol Reachable NLRI (MP_REACH_NLRI)
  - Multiprotocol Unreachable NLRI (MP_UNREACH_NLRI)
- **MP_REACH_NLRI is used**
  - to carry the set of reachable destinations together with the next hop information to be used for forwarding to these destinations
- **MP_UNREACH_NLRI is used**
  - to carry the set of unreachable destinations
- **Both of these attributes**
  - are optional and non-transitive

BGP Update Message Format for IPv4



BGP Update Message Details for IPv4

- **NLRI**
  - 2-tuples of (length, prefix)
    - length = number of masking bits (1 octet)
    - prefix = IP address prefix (1 - 4 octets)
    - note: prefix field contains only necessary bits to completely specify the IP address followed by enough trailing bits to make the end of the field fall on an octet boundary
- **path attributes are composed of**
  - triples of (type, length, value) -> TLV notation
    - attribute type (two octets)
      - 8 bit attribute flags, 8 bit attribute type code
    - attribute length (one or two octets)
      - signaled by attribute flag-bit nr.4
    - attribute value (variable length)
      - content depends on meaning signaled by attribute type code

IPv4 Path Attribute Format / NLRI Format

1 octet	Attribute Type 1 octet	1-2 octet
Attr. Flags	Attr. Type Code	Attribute Length
Attribute Value (variable octets)		

Path Attribute Format

1 octet	
Length	Prefix (1 - 4 octets)

NLRI

VPN-IPv4 BGP Update with MP_Reach_NLRI

1 octet	1 octet	1 octet
Attr. Flags	Type Code = 14	Attribute Length
AFI= 1		SAFI = 128
Length of NHA		
Next Hop Address (NHA, 1- 4 octets)		

Path Attribute MP_Reach_NLRI

1 octet
Length = 120
Label (3 octets = 24 bits)
Route Distinguisher (8 octets = 64 bits)
IPv4 address (4 octets = 32 bits)

NLRI for VPN-IPv4

Format of Attribute-Type

- **8 bit attribute flags**
  - 1. bit (MSB)
    - optional (1) or well-known (0)
  - 2. bit
    - transitive (1) or non-transitive (0)
    - only for optional; set to 1 for well-known
  - 3. bit
    - partial (1) or complete (0)
    - set to 0 for well-known and optional non-transitive
  - 4. bit
    - two octet (1) or one octet (0) attribute length field
- **8 bit attribute type code**
  - values 1 - 16 currently defined

Classification of Attributes 1

- **well-known**
  - must be recognized by all BGP implementations
- **well-known mandatory**
  - must be included in every Update message
    - Origin, AS_Path, Next_Hop
- **well-known discretionary**
  - may or may not be included in every Update message
    - Local_Preference, Atomic_Aggregate
- **all well-known attributes must be passed along to other BGP peers**
  - some will be updated properly first, if necessary

Classification of Attributes 2

- **optional**
  - it is not required or expected that all BGP implementation support all optional attributes
  - may be added by the originator or any AS along the path
  - paths are accepted regardless whether the BGP peer understands an optional attribute or not
- **handling of recognized optional attributes**
  - propagation of attribute depends on meaning of the attribute
  - propagation of attribute is not constrained by transitive bit of attribute flags
    - but depends on the meaning of the attribute

Classification of Attributes 3

- **handling of unrecognized optional attribute**
  - propagation of attribute depends on transitive bit of attribute flags
  - transitive
    - paths are accepted (attribute is ignored) and attribute remains unchanged when path is passed along to other peers
    - attribute is marked as partial (bit 3 of attribute flags)
    - example: Community
  - non-transitive
    - paths are accepted, attribute is quietly ignored and discarded when path is passed along to other peers
    - example: Multi_Exit_Discriminator

Currently Defined Attributes 1

- **Basic attributes**
  - defined in RFC 1771 (Draft Standard)
  - Origin
    - well-known mandatory; type 1
  - AS_Path
    - well-known mandatory; type 2
  - Next_Hop
    - well-known mandatory; type 3
  - Multi_Exit_Discriminator MED
    - optional non-transitive; type 4
  - Local_Preference
    - well-known discretionary; type 5

Currently Defined Attributes 2

- **Basic attributes (cont.)**
  - Atomic_Aggregate
    - well-known discretionary; type 6
  - Aggregator
    - optional transitive; type 7
  - these are the attributes that you can rely on in a multi-vendor environment

Currently Defined Attributes 3

- **Advanced attributes**
  - Community
    - optional transitive; type 8
    - defined in RFC 1997 (Proposed Standard)
  - Originator_ID
    - optional non-transitive; type 9
    - defined in RFC 1966 (Experimental) and RFC 2796 (Proposed Standard) -> Route Reflector
  - Cluster_List
    - optional non-transitive; type 10
    - defined in RFC 1966 (Experimental) and RFC 2796 (Proposed Standard) -> Route Reflector

Currently Defined Attributes 4

- **Advanced attributes (cont.)**
  - Multiprotocol Reachable NLRI
    - MP_REACH_NLRI
    - optional non-transitive; type 14
    - defined in RFC 2858 (Proposed Standard) -> Multiprotocol Extensions
  - Multiprotocol Unreachable NLRI
    - MP_UNREACH_NLRI
    - optional non-transitive; type 15
    - defined in RFC 2858 (Proposed Standard) -> Multiprotocol Extensions
  - in a multi-vendor environment carefully check implementation details

Community Attribute Review 1

- **optional transitive attribute**
- **community is a group of destinations that share a common property**
  - group of networks which should be handled by a foreign AS in a certain way
  - community is not restricted to one network or one AS
- **community attributes are used**
  - to simplify routing policy based on logical properties rather than IP prefix or AS number (= physical location)
  - to tag routes to ensure consistent filtering or route-selection policy

 MPLS_VPN, v4.6 21

Community Attribute Review 2

- **32 bit values (range 0 - 4.294.967.200)**
- **well-known communities**
  - 0xFFFFFF01 … No_Export
  - 0xFFFFFF02 … No_Advertise
- **private communities**
  - value range 0x00010000 to 0xFFFEFFFF
  - common practice for using private communities:
    - high order 16 bit: number of AS
      - which is responsible for defining the meaning of the community
    - low order 16 bit: definition of meaning
      - might have only local significance within the defining AS

 MPLS_VPN, v4.6 22

BGP Draft Attributes 1

- **BGP Extended Communities Attribute**
  - consists of a set of "extended communities"
    - optional transitive; type 16
    - defined in draft-ietf-idr-bgp-ext-communities-07.txt
  - two important enhancements over the existing BGP Community Attribute:
    - it provides an extended range, ensuring that communities can be assigned for a plethora of uses, without fear of overlap.
    - the addition of a type field provides structure for the community space.
  - Important for MPLS_VPN
    - **Route Target Community**
    - **Route Origin Community**

BGP Draft Attributes 2

- **Route Target:**
  - The Route Target Community identifies one or more routers that may receive a set of routes (that carry this Community) carried by BGP. This is transitive across the Autonomous system boundary.
  - It really identifies only a set of sites which will be able to use the route, without prejudice to whether those sites constitute what might intuitively be called a VPN.
- **Route Origin:**
  - The Route Origin Community identifies one or more routers that inject a set of routes (that carry this Community) into BGP. This is transitive across the Autonomous system boundary.

BGP Draft Attributes 3

- **Route Target and Router Origin**
  - type: 2 octets (extended form of this attribute)
    - high octet -> 00, 01, 02 -> defines the structure of the value field
    - low octet -> defines the actual type
  - value: 6 octets
- **Route Target:**
  - high octet type: 0x00 or 0x01 or 0x02
  - low octet type: 0x02
- **Route Origin:**
  - high octet type: 0x00 or 0x01 or 0x02
  - low octet type: 0x03

BGP Draft Attributes 4

- **Structure of value field based on high octet part of type**
  - 0x00:
    - 2 octets Global Administrator Field (IANA assigned AS #)
    - 4 octets Local Administrator Field (actual value of given type contained in low octet part of type)
  - 0x01:
    - 4 octets Global Administrator Field (IP address assigned by IANA)
    - 2 octets Local Administrator Field
  - 0x02:
    - 4 octets Global Administrator Field (IANA assigned 4 octet AS #)
    - 2 octets Local Administrator Field

Agenda

- MP-BGP
- VPN Overview
- MPLS VPN Architecture
- MPLS VPN Basic VPNs
- MPLS VPN Complex VPNs
- MPLS VPN Configuration (Cisco)
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

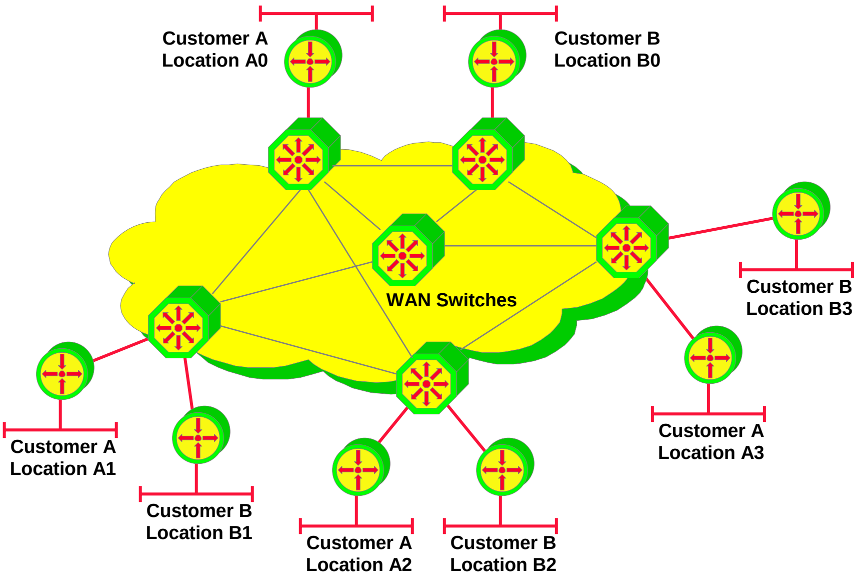
Classical VPN´s

- X.25, Frame Relay or ATM in the core
- dedicated physical switch ports for every customers CPE
  - router, bridge, computer
- customer traffic separation in the core done by concept of virtual circuit
  - PVC service
    - management overhead
  - SVC service with closed user group feature
    - signaling overhead
- separation of customers inherent to virtual circuit technique
- privacy is aspect of customer
  - in most cases overlooked

VPN's based on Overlay Model

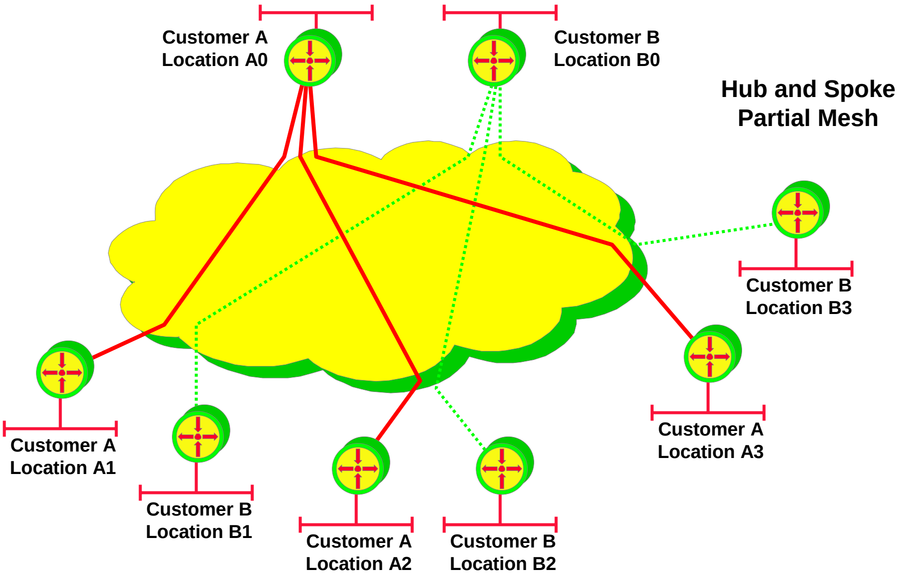
L86 - MPLS VPN

Physical Topology of Classical VPN



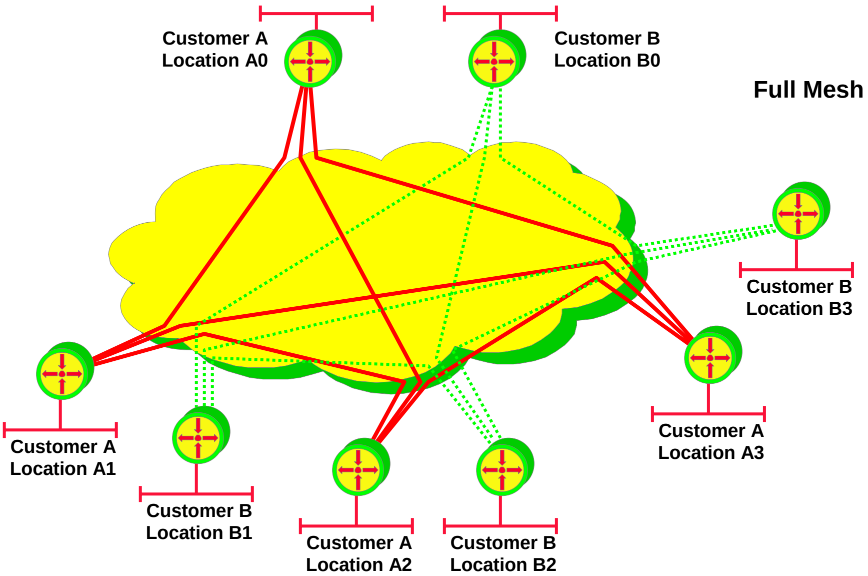
© 2006, D.I. Manfred Lindner MPLS_VPN, v4.6 29

Logical Topology Classic VPN (1)



© 2006, D.I. Manfred Lindner MPLS_VPN, v4.6 30

Logical Topology Classic VPN (2)

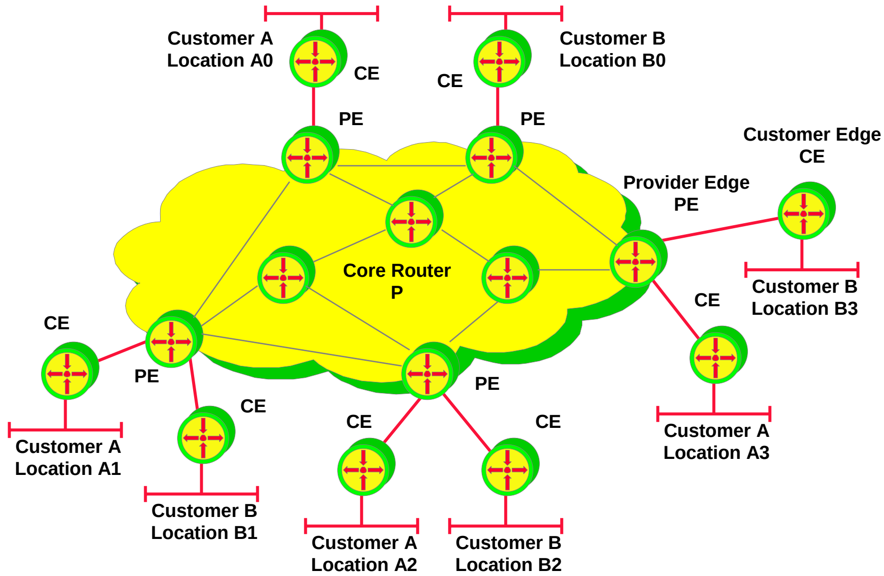


Virtual Private Networks based on IP

- single technology end-to-end
  - IP forwarding and IP routing
- no WAN switches in the core
  - based on different technology (X.25, FR or ATM)
  - administered by different management techniques
- but accounting and quality of service just coming in the IP world
  - X.25, FR and ATM have it already
- often private means cases control over separation but not privacy
  - data are seen in clear-text in the core
  - encryption techniques can solve this problem
  - but encryption means must be in the hand of the customer

VPN´s based on Peer Model

Physical Topology IP VPN



 MPLS_VPN, v4.6 33

Possible Solutions for IP VPN´s

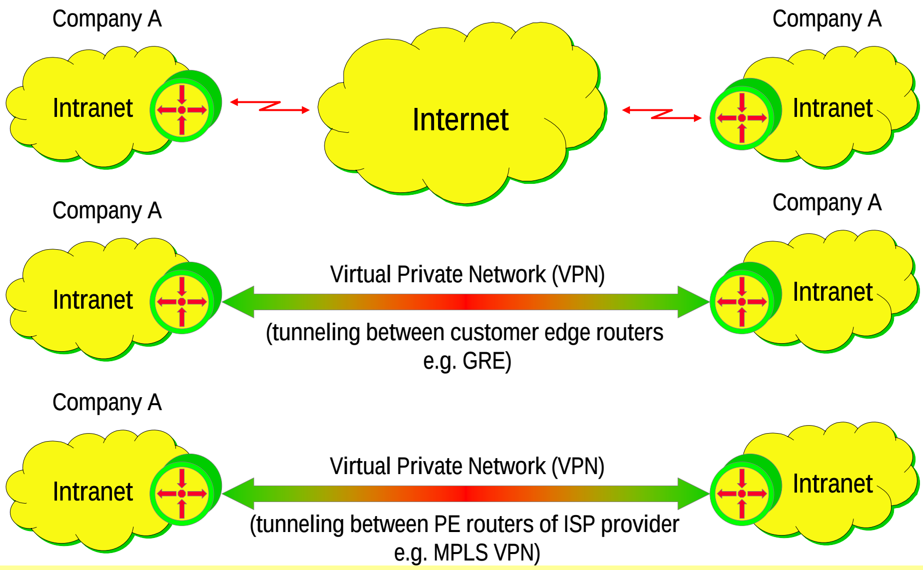
- **IP addresses of customers non overlapping**
  - filtering and policy routing techniques can be used in order to guarantee separation of IP traffic
    - exact technique depends on who manages routes at the customer site
- **IP addresses of customers overlapping**
  - tunneling techniques must be used in order to guarantee separation of IP traffic
    - GRE
    - L2F, PPTP, L2TP
    - MPLS-VPN
- **If privacy is a topic**
  - encryption techniques must be used
    - SSL/TLS, IPsec

 MPLS_VPN, v4.6 34

Tunneling Solutions for IP VPN´s

- **Tunneling techniques are used in order to guarantee separation of IP traffic**
  - IP in IP Tunneling or GRE (Generic Routing Encapsulations)
    - Bad performance on PE router
  - PPTP or L2TP for LAN to LAN interconnection
    - Originally designed for PPP Dial-up connections
    - LAN – LAN is just a special case
  - MPLS-VPN
    - Best performance on PE router
- **In all these cases**
  - Privacy still an aspect of the customer

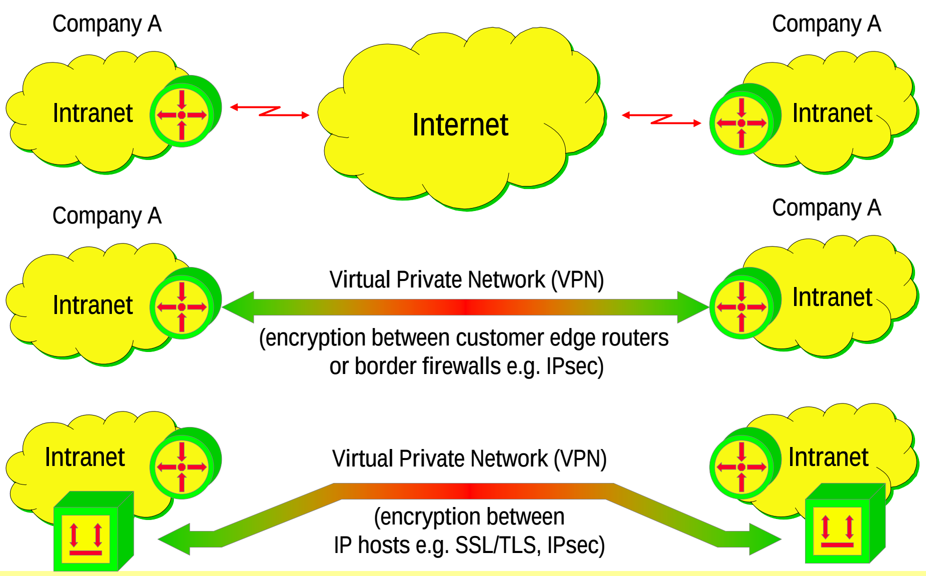
Tunneling IP VPNs without Encryption



Encryption Solutions for IP VPN´s

- **If privacy is a topic tunneling techniques with encryption are used in order to hide IP traffic**
  - SSL (secure socket layer)
    - Usually end-to-end
    - Between TCP and Application Layer
  - IPsec
    - Could be end-to-end
    - Could be between special network components (e.g. firewalls, VPN concentrators) only
    - Between IP and TCP/UDP Layer
  - PPTP and L2TP Tunnels
    - With encryption turned on via PPP option

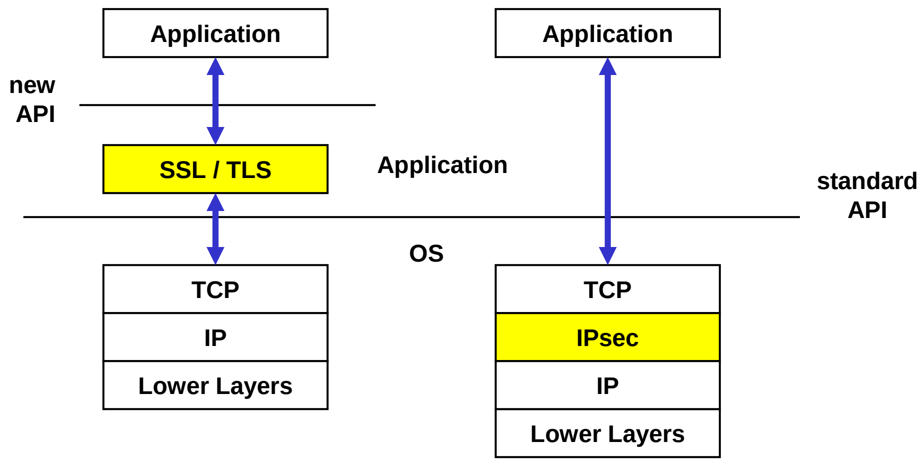
Tunneling IP VPNs without Encryption



SSL/TLS versus IPsec

Application must be aware of new application programming interface

Application can use standard application programming interface



Two Major VPN Paradigms

- **Overlay VPNs: Transparent P2P links**
  – Well-known technology
  – Provider does not care about customer routing
  – Best customer isolation
- **Peer VPNs: Participation in P-routing**
  – Optimum routing
  – Simple provision of additional VPN
  – Problems with address space

MPLS VPN – Best of Both Worlds

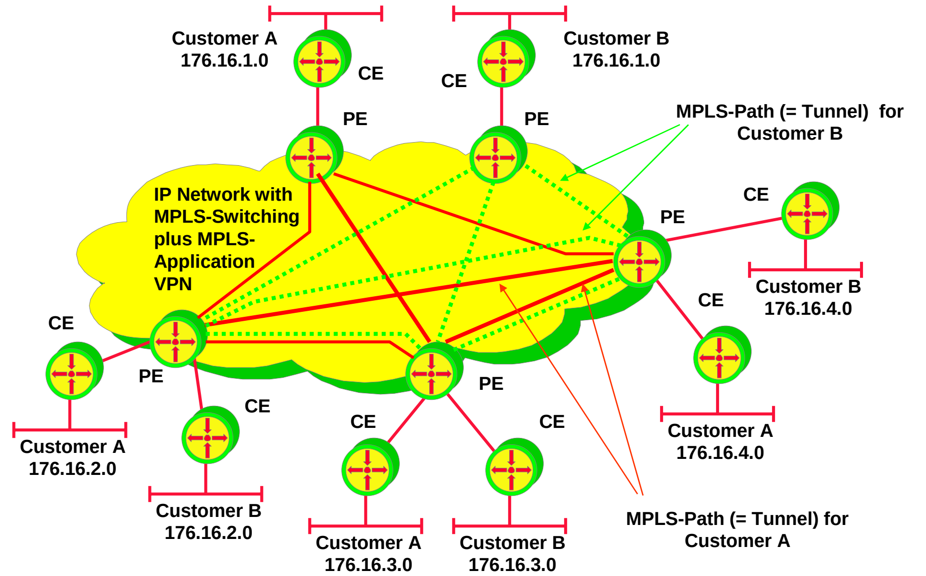
- **Combines VPN Overlay model with VPN Peer model**
- **PE routers allow route isolation**
  - By using Virtual Routing and Forwarding Tables (VRF) for differentiating routes from the customers
  - Allows overlapping address spaces
- **PE routers participate in P-routing**
  - Hence optimum routing between sites
  - Label Switches Paths are used within the core network
  - Easy provisioning (sites only)
- **Overlapping VPNs possible**
  - By a simple (?) attribute syntax

MPLS VPN – Principles

- **Requires MPLS Transport within the core**
  - Using the label stack feature of MPLS
- **Requires MP-BGP among PE routers**
  - Supports IPv4/v6, VPN-IPv4, multicast
  - Default behavior: BGP-4
- **Requires VPN-IPv4 96 bit addresses**
  - 64 bit Route Distinguisher (RD)
  - 32 bit IP address
- **Every PE router uses one VRF for each VPN**
  - Virtual Routing and Forwarding Table (VRF)

L86 - MPLS VPN

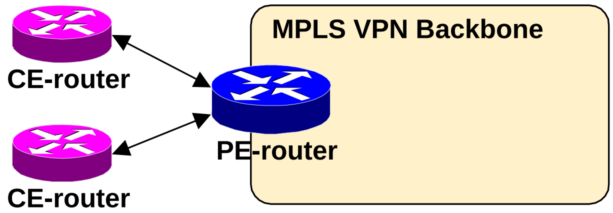
MPLS-VPN



Agenda

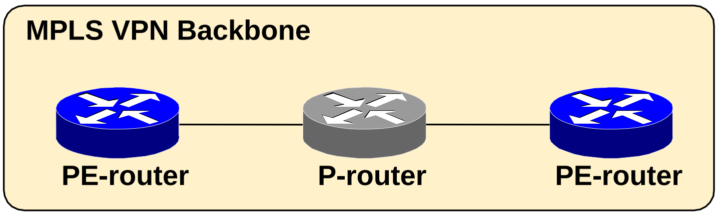
- MP-BGP
- VPN Overview
- <u>MPLS VPN Architecture</u>
- MPLS VPN Basic VPNs
- MPLS VPN Complex VPNs
- MPLS VPN Configuration (Cisco)
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

CE-Router Perspective



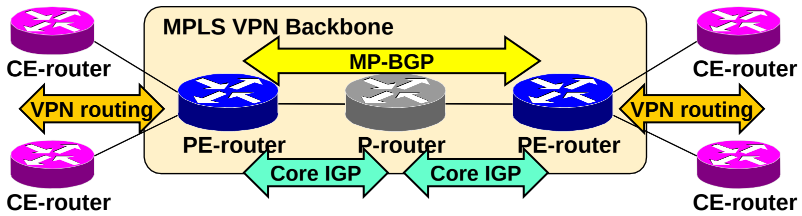
- CE (Customer Edge) - routers run standard IP routing software and exchange routing updates with the PE-router
  - EBGP, OSPF, RIPv2 or static routes are supported
- PE (Provider Edge) - router appears as just another router in the customer's network

P-Router Perspective



- P (Provider) - routers do not participate in MPLS VPN routing and do not carry VPN (customer) routes
- P - routers run backbone IGP like OSPF or IS-IS with the PE-routers

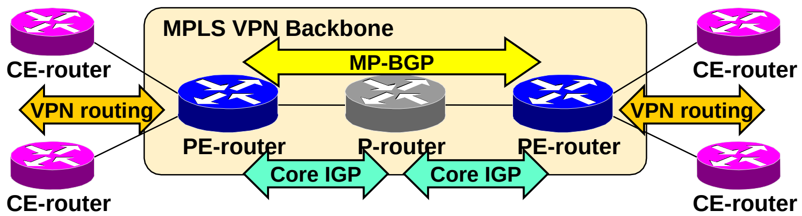
PE-Router Perspective



PE-routers contain a number of routing tables:

- Global routing table that contains core routes (filled with core IGP)
- Virtual Routing and Forwarding (VRF) tables for sets of sites with identical routing requirements
- VRF's are filled with information from CE-routers and MP-BGP information from other PE-routers

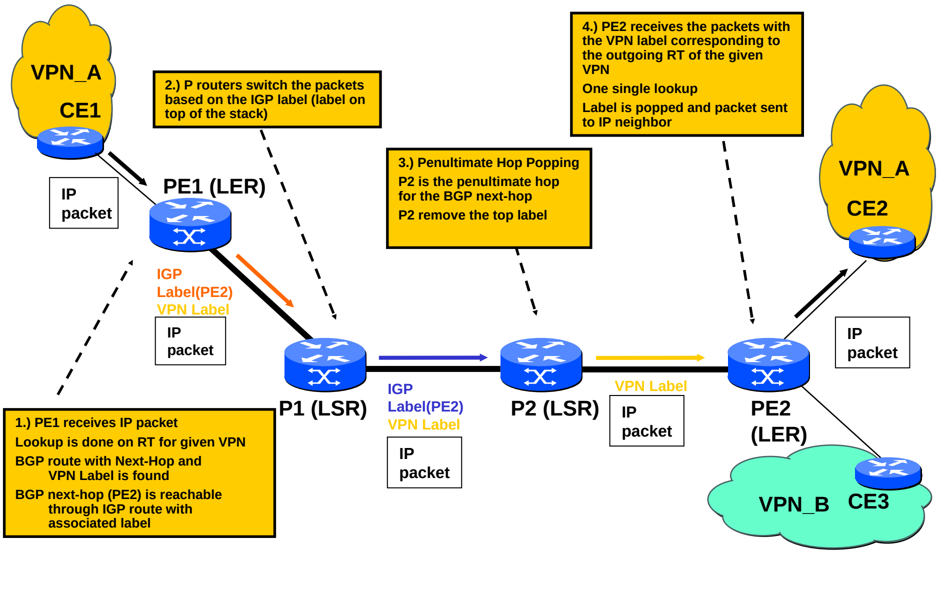
PE-Router Perspective



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 VPN-IPv4 routes with other PE-routers via Internal MP-BGP sessions

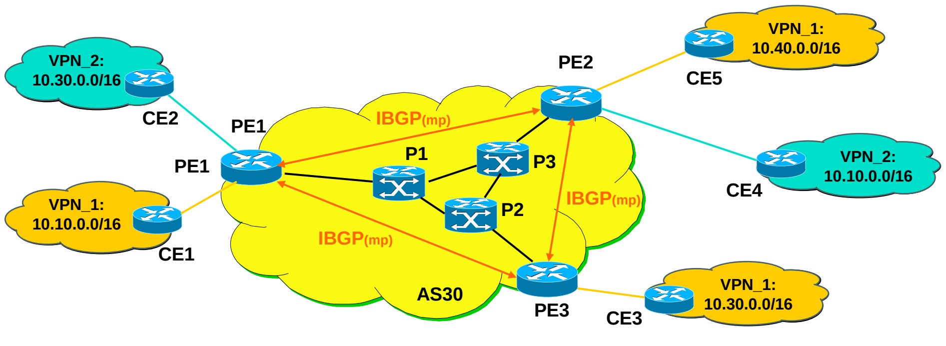
MPLS VPN using MPLS Label Stack



Agenda

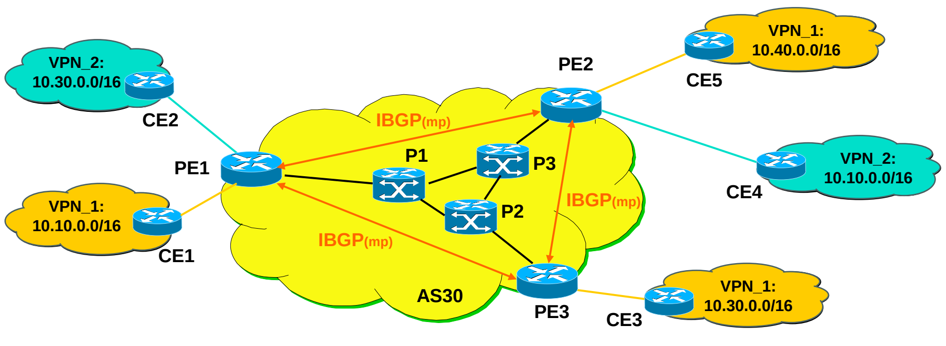
- MP-BGP
- VPN Overview
- MPLS VPN Architecture
- MPLS VPN Basic VPNs
- MPLS VPN Complex VPNs
- MPLS VPN Configuration (Cisco)
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

VPN MPLS Architecture 1



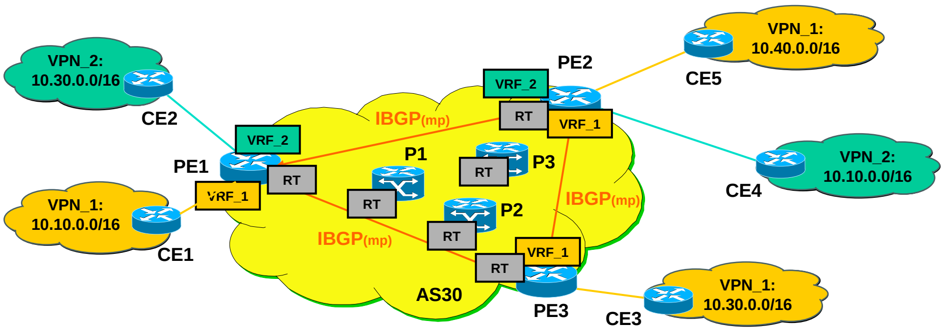
- Customer Networks are connected to MPLS VPN service provider
- Basic scenario:
  - VPN's are not overlapping -> that means they are totally separated from each other
  - VPN_1: Orange Customer
  - VPN_2: Green Customer

VPN MPLS Architecture 2



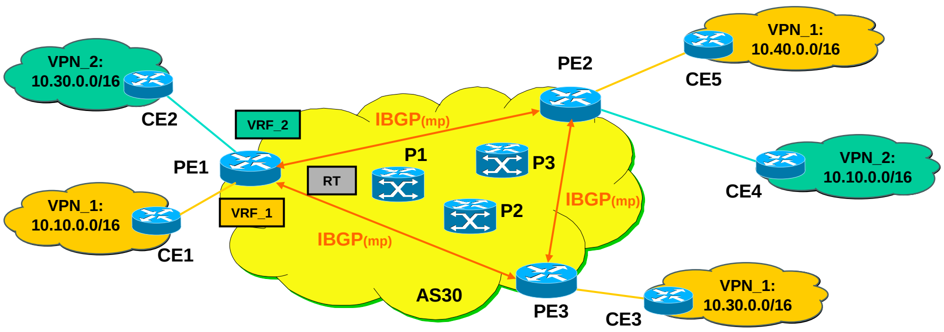
- Provider routers P (LSR's) are in the core of the MPLS cloud
- Provider Edge routers PE (LER) use MPLS within the core and plain IP with CE routers
- PE routers are fully meshed concerning Internal MP-BGP Sessions
- P and PE routers share a common IGP (e.g. OSPF or IS-IS)
- Customer Edge CE routers connect customer sites to provider

VPN MPLS Architecture 3



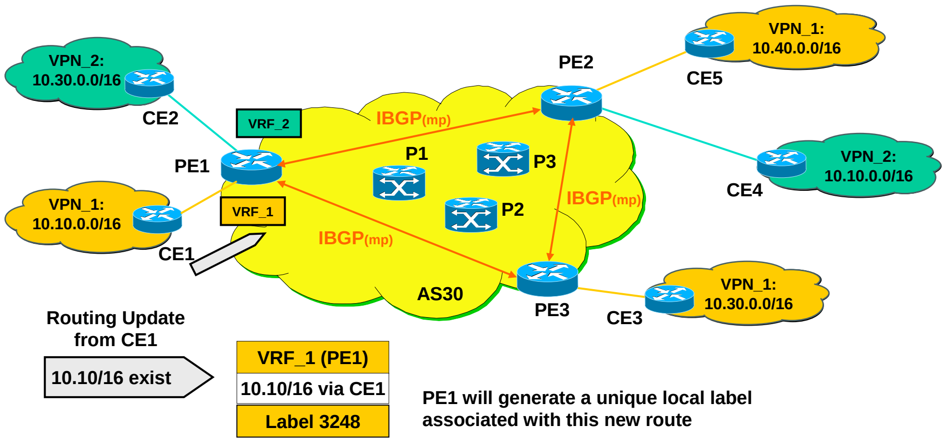
- PE router
  - maintains a separate routing table VRF per customer site
    - VRF (VPN Routing and Forwarding) Table
  - holds global routing table RT for communication within MPLS cloud
    - maintained by IGP
  - forwarding within MPLS cloud is based on labels
    - distributed by LDP

VPN MPLS Architecture 4



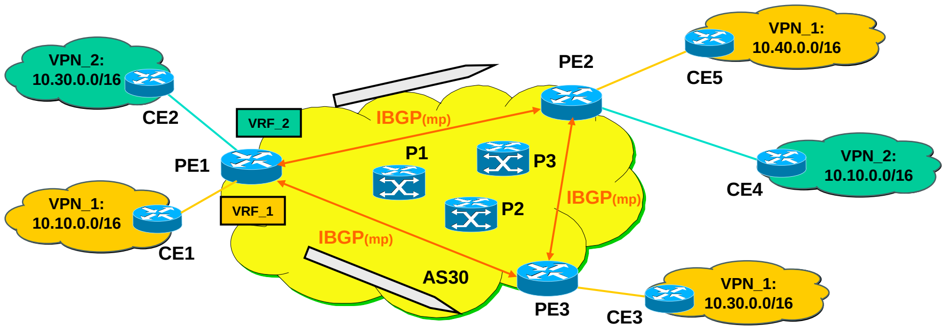
- VRF table
  - contains Net-IDs received from corresponding CE site
    - via RIPv2, OSPF, External BGP session or static routes
  - contains NET-IDs received from other PE routers
    - via Internal MP-BGP Sessions received as VPN-IPv4 addresses
  - hence overlapping addresses are no problem

New Network 10.10.0.0/16 at CE1 1



- Routing Update will install a new route in the corresponding VRF table of PE1 and hence the new route must be advertised to all other PE´s via Internal MP-BGP
  - as VPN-IPv4 address

Advertise Network 10.10.0.0/16 to PE’s 2



MP-BGP uses:

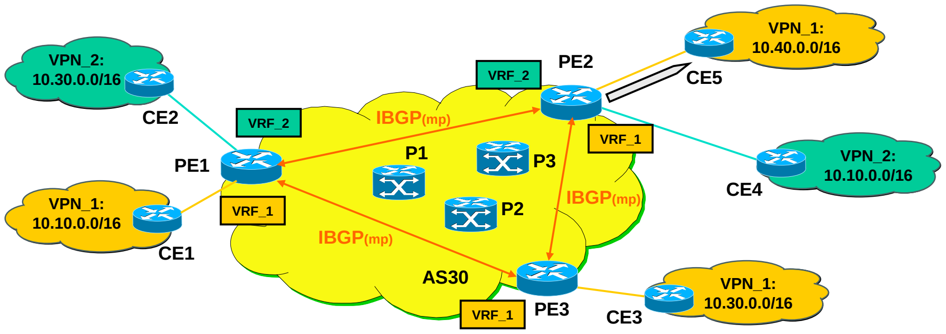
MP_Reach_NLRI attribute
Next-Hop
VPN-IPv4_NLRI
RD=Route Distinguisher
Net
Label
Extended Community attr.
RT = Route Target

Routing Update from PE1 via Internal MP-BGP to all other PE´s

AS30 VPN #1

VPN-IPv4 update:
RD (ID to uniquely distinguished Net from other nets) = 30:1
Net = 10.10/16, Next-Hop = PE1
Label that should be used to reach this Net = 3248
RT (Hint to which VRF´s this Net should be imported) = Orange

New Network 10.10.0.0/16 at PE2/CE5 3



Routing Update from PE1
received at PE 2

VPN-IPv4 update:
RD = 30:1
Net = 10.10/16, Next-Hop = PE1
Label = 3248
RT = Orange

New Route put into
VRF_1 based on RT=Orange

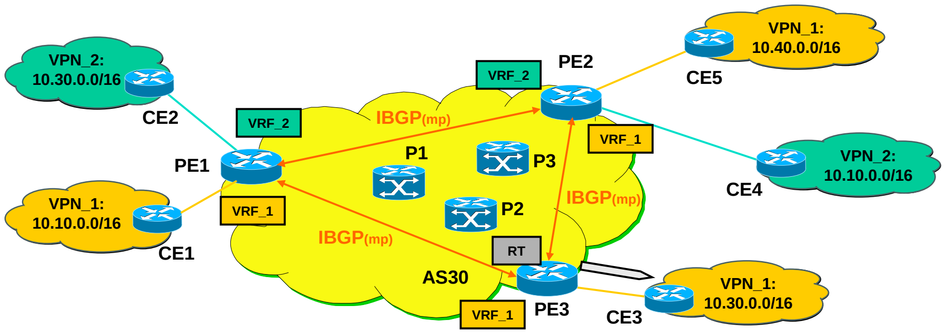
VRF_1 (PE2)
10.10/16 via PE1 use 3248

VRF_2 (PE2)

Routing Update
to CE5

10.10/16 exist

New Network 10.10.0.0/16 at PE3/CE3 4



Routing Update from PE1
received at PE 3

VPN-IPv4 update:
RD = 30:1
Net = 10.10/16, Next-Hop = PE1
Label = 3248
RT = Orange

New Route put into
VRF_1 based on RT=Orange

VRF_1 (PE3)
10.10/16 via PE1 use 3248

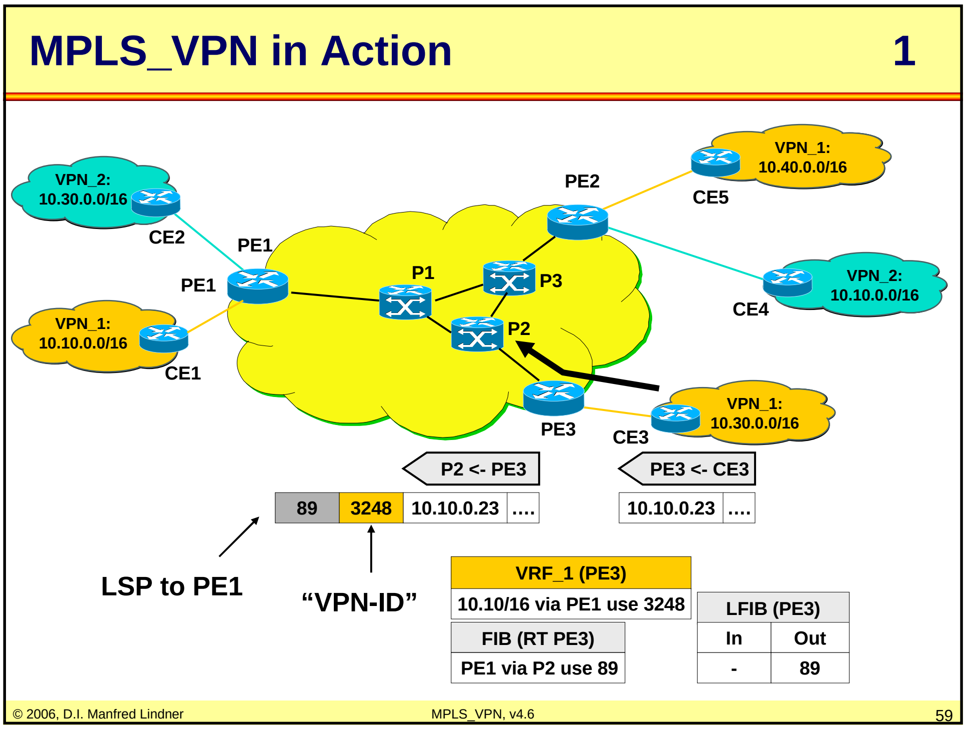
FIB (RT PE3)
PE1 via P2 use 89

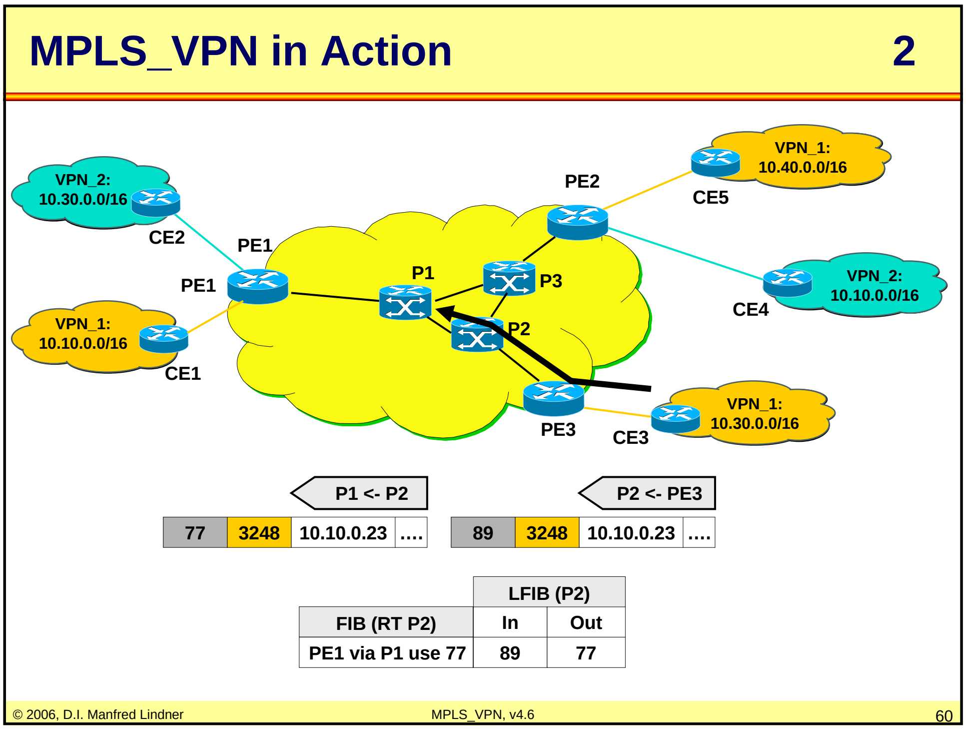
Routing Update
to CE3

10.10/16 exist

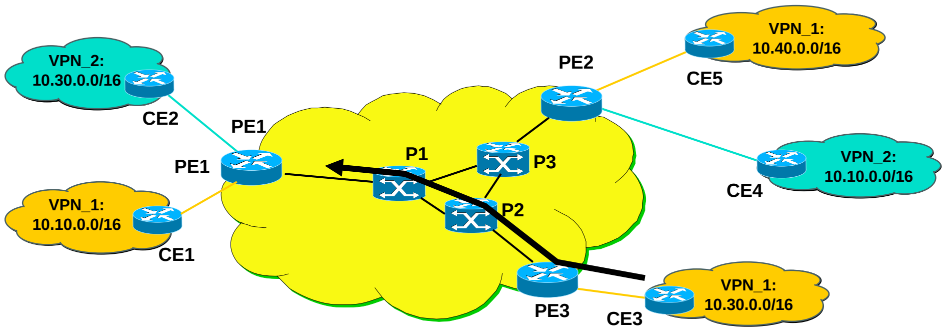
RT (CE3)
10.10/16 via PE3

L86 - MPLS VPN


MPLS_VPN in Action
1
VPN_2: 10.30.0.0/16
CE2
PE1
PE1
VPN_1: 10.10.0.0/16
CE1
P1
P3
P2
PE2
VPN_1: 10.40.0.0/16
CE5
VPN_2: 10.10.0.0/16
CE4
PE3
CE3
VPN_1: 10.30.0.0/16
P2 <- PE3
PE3 <- CE3
89
3248
10.10.0.23
....
10.10.0.23
....
LSP to PE1
"VPN-ID"
VRF_1 (PE3)
10.10/16 via PE1 use 3248
FIB (RT PE3)
PE1 via P2 use 89
LFIB (PE3)
In
Out
-
89
© 2006, D.I. Manfred Lindner
MPLS_VPN, v4.6
59


MPLS_VPN in Action
2
VPN_2: 10.30.0.0/16
CE2
PE1
PE1
VPN_1: 10.10.0.0/16
CE1
P1
P3
P2
PE2
VPN_1: 10.40.0.0/16
CE5
VPN_2: 10.10.0.0/16
CE4
PE3
CE3
VPN_1: 10.30.0.0/16
P1 <- P2
P2 <- PE3
77
3248
10.10.0.23
....
89
3248
10.10.0.23
....
LFIB (P2)
FIB (RT P2)
In
Out
PE1 via P1 use 77
89
77
© 2006, D.I. Manfred Lindner
MPLS_VPN, v4.6
60

MPLS_VPN in Action 3



PE1 <- P1

3248	10.10.0.23	....

P1 <- P2

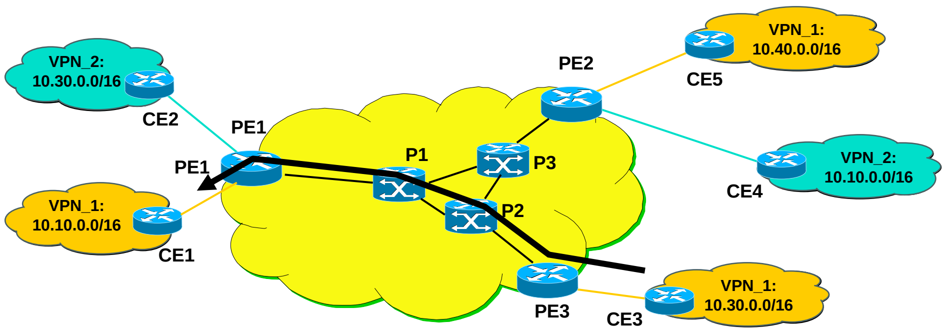
77	3248	10.10.0.23	....

FIB (RT P1)
PE1 via PE1 use null

LFIB (P1)	
In	Out
77	POP

 MPLS_VPN, v4.6 61

MPLS_VPN in Action 4



PE1 <- CE1

10.10.0.23	....

PE1 <- P1

3248	10.10.0.23	....

VRF_1 (PE1)
10.10/16 via CE1
Label 3248

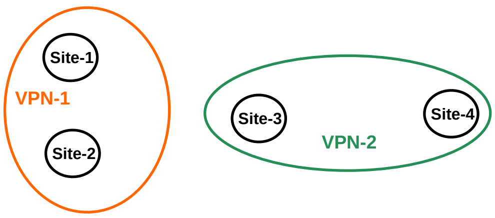
 MPLS_VPN, v4.6 62

Agenda

- MP-BGP
- VPN Overview
- MPLS VPN Architecture
- MPLS VPN Basic VPNs
- MPLS VPN Complex VPNs
- MPLS VPN Configuration (Cisco)
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

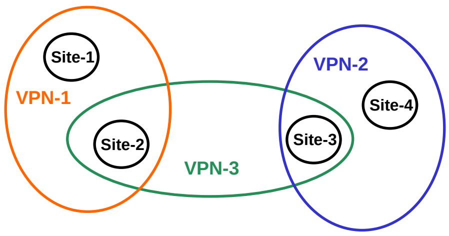
MPLS VPN Types

VPN's not overlapping (Intranet)



site-2 networks can reach site-1 networks and vice versa, site-3 networks can reach site-4 networks and vice versa.

VPN's overlapping (Intranet/Extranet)



site-2 networks can reach site-1 and site-3 networks, site-3 networks can reach site-4 and site-3 networks, site-1 networks can reach site-2 networks only, site-4 networks can reach site-3 networks only.

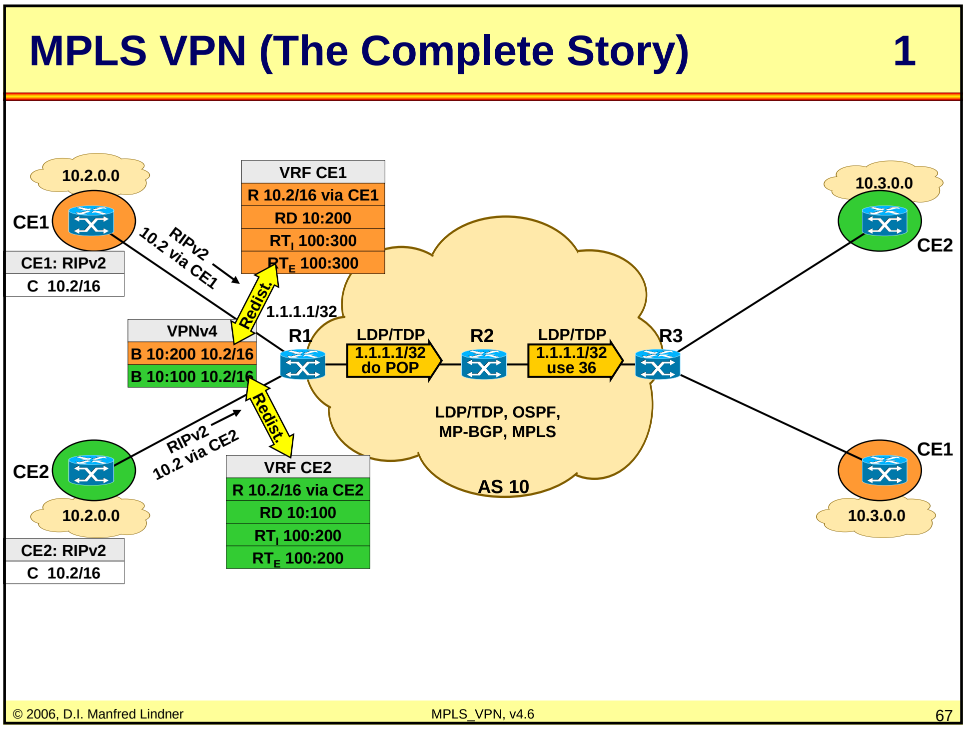
A New Sight of VPN

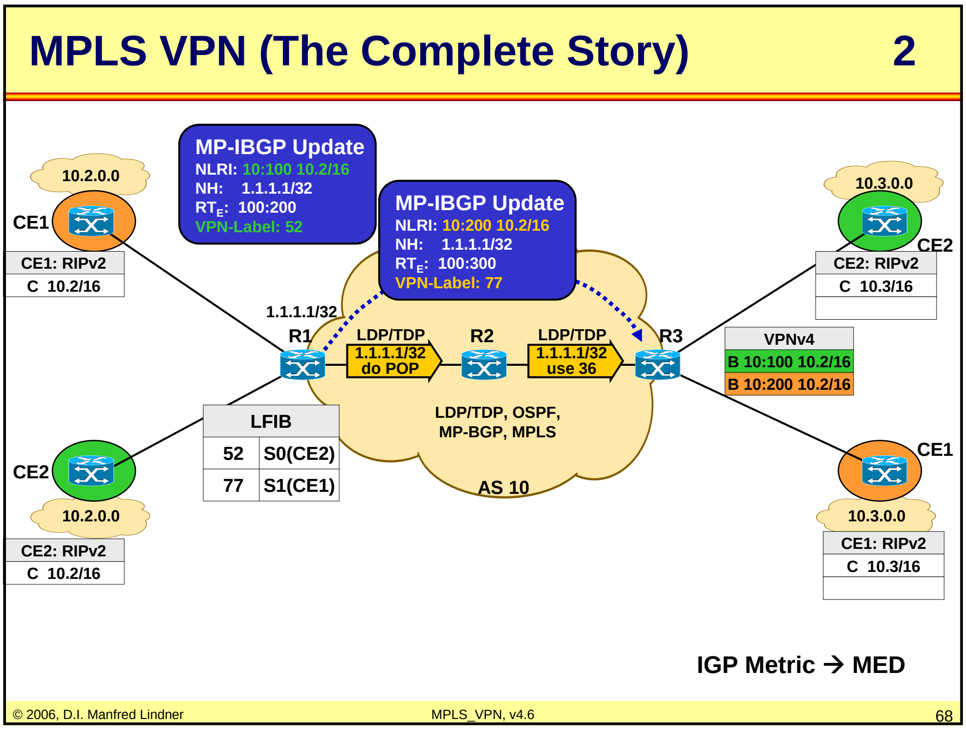
- **For non-overlapping VPN's**
  - The Route Distinguisher would be sufficient
- **For overlapping VPN's**
  - The Route Distinguisher is not sufficient to achieve the new sight (the Extranet policy) of VPNs
- **In order to implement this new sight of VPN´s in case of overlapping VPN's**
  - the Route Target was introduced in the MPLS_VPN Architecture

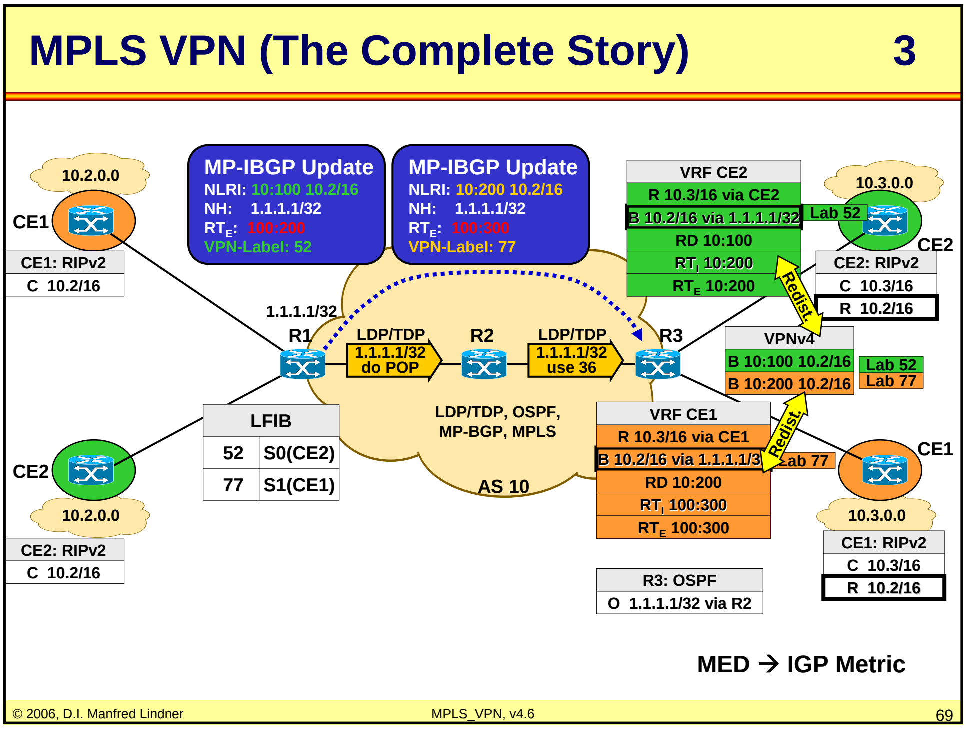
The real Role of the Route Target

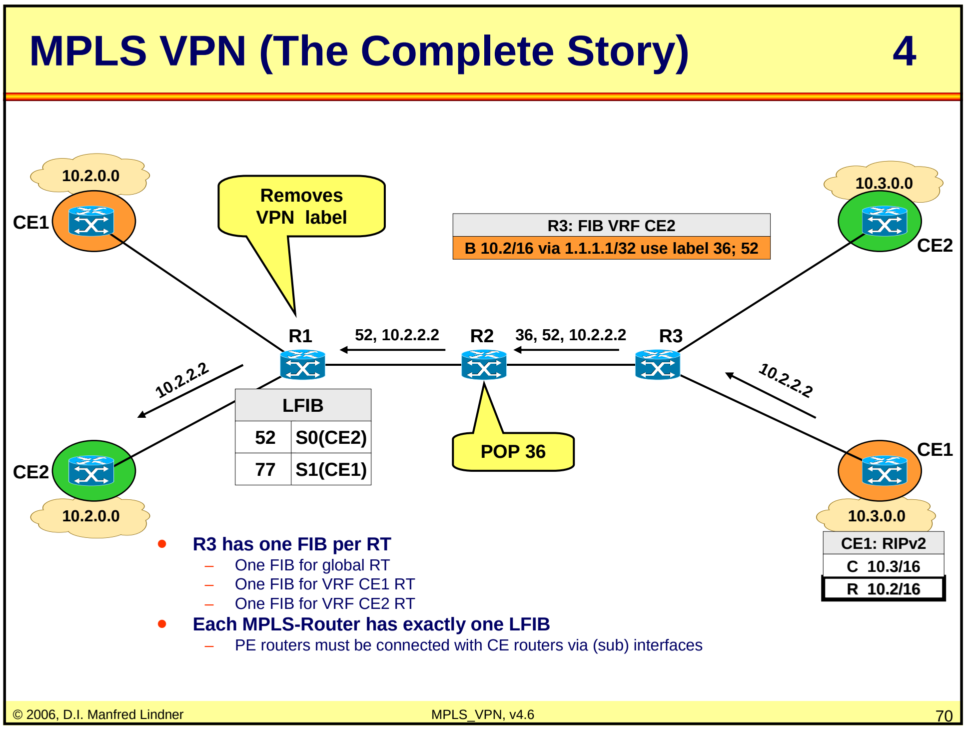
- **PE router which announces a VPNv4 route**
  - uses the Route Target community to specify in which foreign VRF's the announced route should be installed
  - Route Target has export meaning
- **PE router which receives a VPNv4 route**
  - uses the received Route Target community to decide in which local VRF's the announced route should be installed
  - Route Target has import meaning

L86 - MPLS VPN

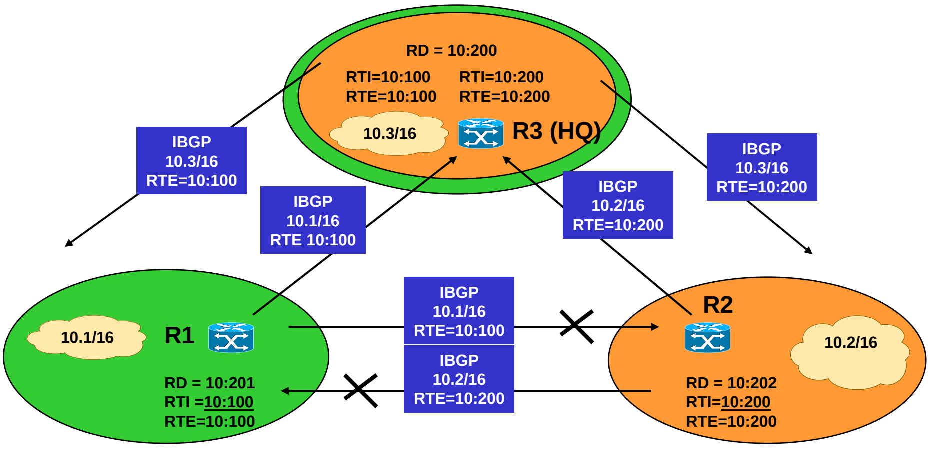

MPLS VPN (The Complete Story) 1
10.2.0.0
CE1
CE1: RIPv2
C 10.2/16
RIPv2
10.2 via CE1
VRF CE1
R 10.2/16 via CE1
RD 10:200
RT$_I$ 100:300
RT$_E$ 100:300
Redist.
1.1.1.1/32
VPNv4
B 10:200 10.2/16
B 10:100 10.2/16
R1
LDP/TDP
1.1.1.1/32 do POP
R2
LDP/TDP
1.1.1.1/32 use 36
R3
LDP/TDP, OSPF, MP-BGP, MPLS
AS 10
RIPv2
10.2 via CE2
Redist.
CE2
10.2.0.0
CE2: RIPv2
C 10.2/16
VRF CE2
R 10.2/16 via CE2
RD 10:100
RT$_I$ 100:200
RT$_E$ 100:200
10.3.0.0
CE2
CE1
10.3.0.0
© 2006, D.I. Manfred Lindner
MPLS_VPN, v4.6
67


MPLS VPN (The Complete Story) 2
MP-IBGP Update
NLRI: 10:100 10.2/16
NH: 1.1.1.1/32
RT$_E$: 100:200
VPN-Label: 52
MP-IBGP Update
NLRI: 10:200 10.2/16
NH: 1.1.1.1/32
RT$_E$: 100:300
VPN-Label: 77
10.2.0.0
CE1
CE1: RIPv2
C 10.2/16
1.1.1.1/32
R1
LDP/TDP
1.1.1.1/32 do POP
R2
LDP/TDP
1.1.1.1/32 use 36
R3
LFIB
52 S0(CE2)
77 S1(CE1)
LDP/TDP, OSPF, MP-BGP, MPLS
AS 10
CE2
10.2.0.0
CE2: RIPv2
C 10.2/16
10.3.0.0
CE2
CE2: RIPv2
C 10.3/16
VPNv4
B 10:100 10.2/16
B 10:200 10.2/16
CE1
10.3.0.0
CE1: RIPv2
C 10.3/16
IGP Metric → MED
© 2006, D.I. Manfred Lindner
MPLS_VPN, v4.6
68


MPLS VPN (The Complete Story) 3
10.2.0.0
CE1
CE1: RIPv2
C 10.2/16
MP-IBGP Update
NLRI: 10:100 10.2/16
NH: 1.1.1.1/32
RT_E: 100:200
VPN-Label: 52
MP-IBGP Update
NLRI: 10:200 10.2/16
NH: 1.1.1.1/32
RT_E: 100:300
VPN-Label: 77
VRF CE2
R 10.3/16 via CE2
B 10.2/16 via 1.1.1.1/32
RD 10:100
RT_I 10:200
RT_E 10:200
Lab 52
10.3.0.0
CE2
CE2: RIPv2
C 10.3/16
R 10.2/16
Redist.
1.1.1.1/32
R1
LDP/TDP
1.1.1.1/32
do POP
R2
LDP/TDP
1.1.1.1/32
use 36
R3
VPNv4
B 10:100 10.2/16
B 10:200 10.2/16
Lab 52
Lab 77
LFIB
52 S0(CE2)
77 S1(CE1)
LDP/TDP, OSPF,
MP-BGP, MPLS
AS 10
VRF CE1
R 10.3/16 via CE1
B 10.2/16 via 1.1.1.1/32
Lab 77
RD 10:200
RT_I 100:300
RT_E 100:300
Redist.
CE1
10.3.0.0
CE1: RIPv2
C 10.3/16
R 10.2/16
CE2
10.2.0.0
CE2: RIPv2
C 10.2/16
R3: OSPF
O 1.1.1.1/32 via R2
MED → IGP Metric
© 2006, D.I. Manfred Lindner
MPLS_VPN, v4.6
69


MPLS VPN (The Complete Story) 4
10.2.0.0
CE1
Removes
VPN label
R3: FIB VRF CE2
B 10.2/16 via 1.1.1.1/32 use label 36; 52
10.3.0.0
CE2
R1
52, 10.2.2.2
R2
36, 52, 10.2.2.2
R3
10.2.2.2
10.2.2.2
LFIB
52 S0(CE2)
77 S1(CE1)
POP 36
CE2
10.2.0.0
CE1
10.3.0.0
CE1: RIPv2
C 10.3/16
R 10.2/16
R3 has one FIB per RT
– One FIB for global RT
– One FIB for VRF CE1 RT
– One FIB for VRF CE2 RT
Each MPLS-Router has exactly one LFIB
– PE routers must be connected with CE routers via (sub) interfaces
© 2006, D.I. Manfred Lindner
MPLS_VPN, v4.6
70

Example for Overlapping VPNs using Different Route Targets

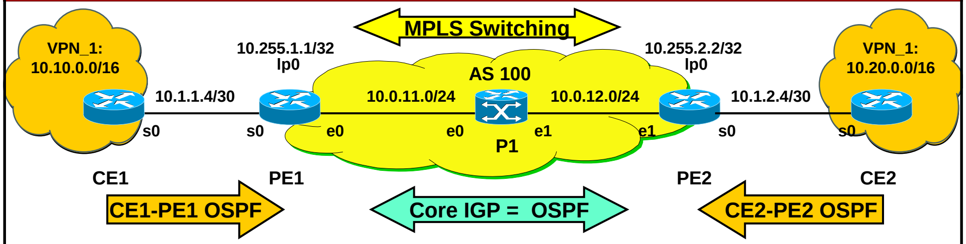


- IBGP Split Horizon Rule assures that R3 (HQ) does not forward routes learned by peers
- IP addresses must be unique in overlapping situations!

Agenda

- MP-BGP
- VPN Overview
- MPLS VPN Architecture
- MPLS VPN Basic VPNs
- MPLS VPN Complex VPNs
- MPLS VPN Configuration (Cisco)
  – CE-PE OSPF Routing
  – CE-PE Static Routing
  – CE-PE RIP Routing
  – CE-PE External BGP Routing

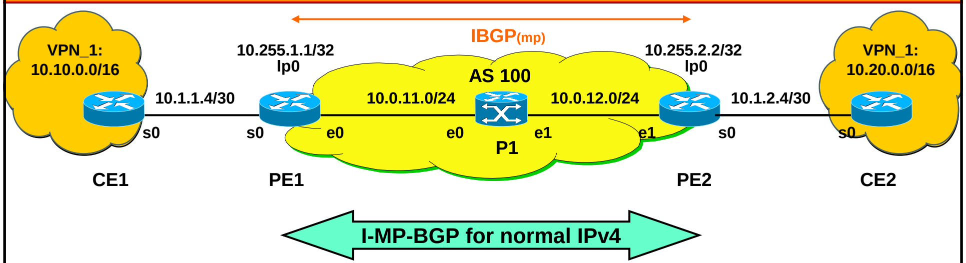
IP Addressing, OSPF Routing in VPN_1, Basic OSPF Routing and MPLS in AS 100




```
CE1:
int s0
  ip address 10.1.1.5 255.255.255.252
router ospf 10
  network 10.1.1.5 0.0.0.0 area 0

PE1:
ip cef
int loopback 0
  ip address 10.255.1.1 255.255.255.255
int s0
  ip address 10.1.1.6 255.255.255.252
int e0
  ip address 10.0.11.1 255.255.255.0
  mpls ip
router ospf 100
 network 10.0.11.1 0.0.0.0 area 0
```

```
CE2:
int s0
  ip address 10.1.2.5 255.255.255.252
router ospf 10
  network 10.1.2.5 0.0.0.0 area 0

PE2:
ip cef
int loopback 0
  ip address 10.255.2.2 255.255.255.255
int s0
  ip address 10.1.2.6 255.255.255.252
int e0
  ip address 10.0.12.2 255.255.255.0
  mpls ip
router ospf 100
 network 10.0.12.1 0.0.0.0 area 0
```

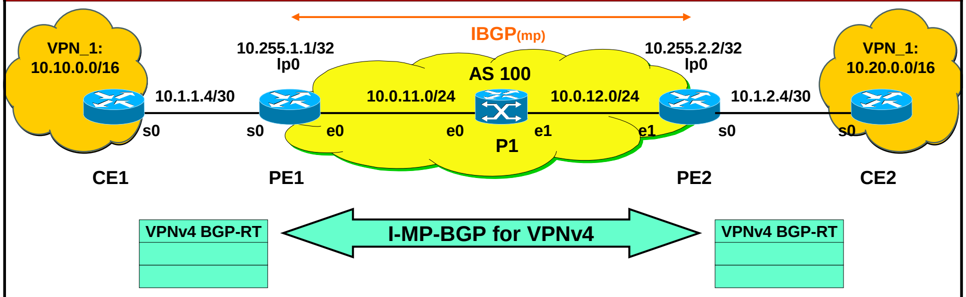

Start Normal I-BGP in AS 100




```
PE1:
int loopback 0
  ip address 10.255.1.1 255.255.255.255

router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
  address-family ipv4
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family
```

```
PE2:
int loopback 0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family ipv4
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family
```

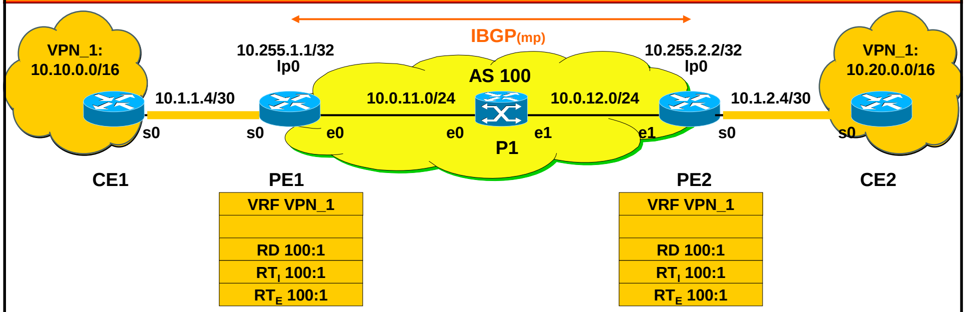
Start MP-BGP in AS 100



PE1:
int loopback 0
  ip address 10.255.1.1 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
  address-family vpnv4
    neighbor 10.255.2.2 activate
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 send-community extended (default)
  exit-address-family

PE2:
int loopback 0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family vpnv4
    neighbor 10.255.1.1 activate
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 send-community extended
  exit-address-family

Create VRF and Bring Interface into VRF (PE router)



PE1:

ip vrf *VPN_1*
  rd 100:1
  route-target import 100:1
  route-target export 100:1

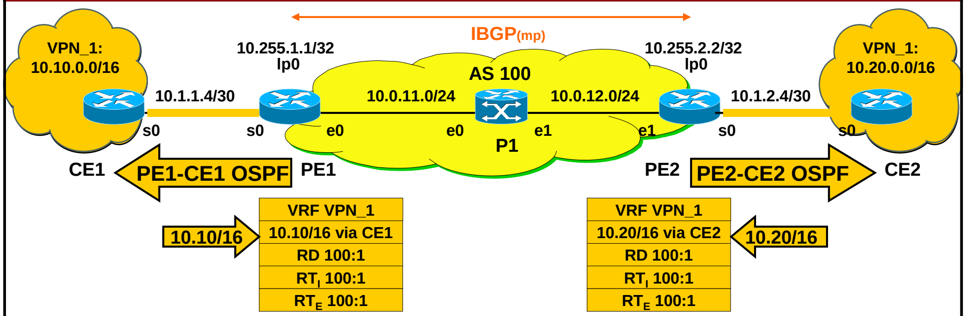
int s0
  ip vrf forwarding *VPN_1*
  ip address 10.1.1.6 255.255.255.252

PE2:

ip vrf *VPN_1*
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding *VPN_1*
  ip address 10.1.2.6 255.255.255.252

Start Dynamic Routing (OSPF) towards CE (PE router)



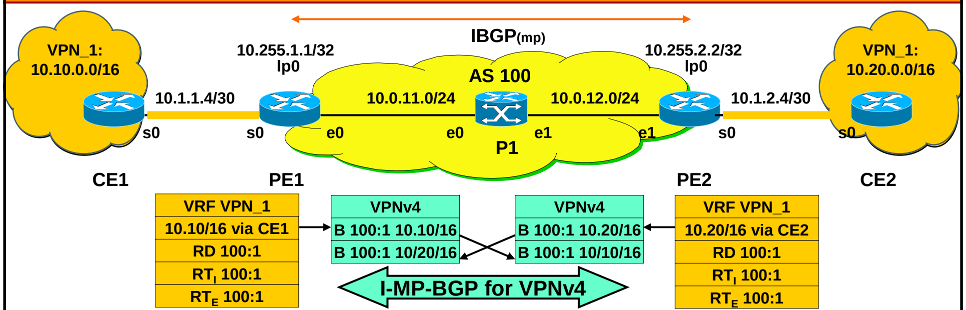
PE1:

```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.1.6 255.255.255.252

router ospf 110 vrf VPN_1
  network 10.1.1.6 0.0.0.0 area 0
```

PE2:

```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.2.6 255.255.255.252

router ospf 120 vrf VPN_1
  network 10.1.2.6 0.0.0.0 area 0
```


Redistributing VRF OSPF into MP-BGP and Transport of VPNv4 routes via I-MP-BGP (PE router)



PE1:

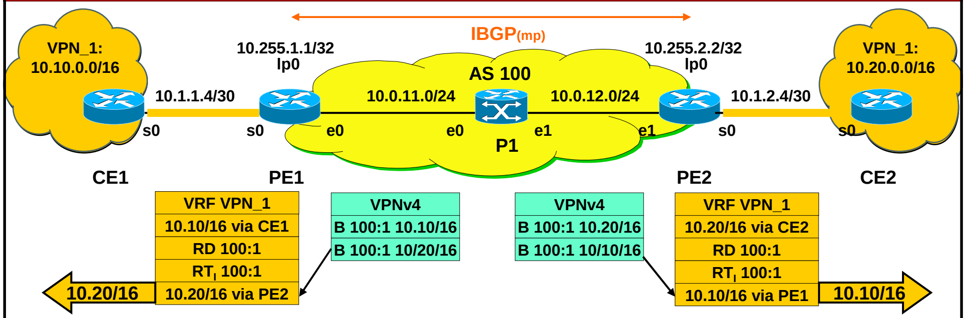
```
ip vrf VPN_1
  rd 100:1

router ospf 110 vrf VPN_1
  network 10.1.1.6 0.0.0.0 area 0

router bgp 100
   no bgp default ipv4-unicast
   address-family ipv4 vrf VPN_1
    redistribute ospf 110 match internal
    no auto-summary
    no synchronization
   exit-address-family
```

PE2:

```
ip vrf VPN_1
  rd 100:1

router ospf 120 vrf VPN_1
  network 10.1.2.6 0.0.0.0 area 0

router bgp 100
   no bgp default ipv4-unicast
   address-family ipv4 vrf VPN_1
    redistribute ospf 120  match internal
    no auto-summary
    no synchronization
   exit-address-family
```


L86 - MPLS VPN

Redistribution of VPNv4 routes into VRF OSPF (PE router)



PE1:
```
router ospf 110 vrf VPN_1
  network 10.1.1.6 0.0.0.0 area 0
  redistribute bgp 100 metric <20> subnet

router bgp 100
  no bgp default ipv4-unicast
  address-family ipv4 vrf VPN_1
    redistribute ospf 100 match internal
    no auto-summary
  exit-address-family
```

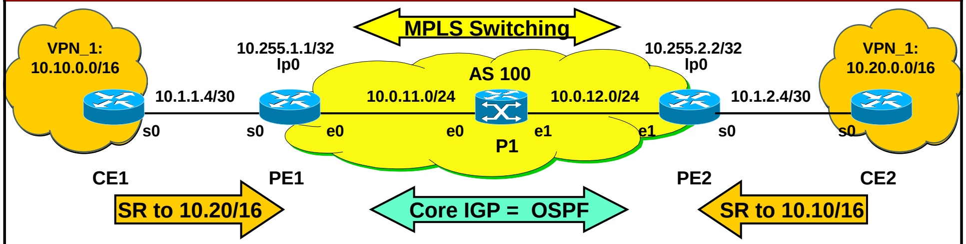
PE2:
```
router ospf 120 vrf VPN_1
  network 10.1.2.6 0.0.0.0 area 0
  redistribute bgp 100 metric <20> subnets

router bgp 100
  no bgp default ipv4-unicast
  address-family ipv4 vrf VPN_1
    redistribute ospf 100 match internal
    no auto-summary
  exit-address-family
```


Agenda

- MP-BGP
- VPN Overview
- MPLS VPN Architecture
- MPLS VPN Basic VPNs
- MPLS VPN Complex VPNs
- MPLS VPN Configuration (Cisco)
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

L86 - MPLS VPN

IP Addressing, Static Routing in VPN_1, Basic OSPF Routing and MPLS in AS 100



CE1:
int s0
  ip address 10.1.1.5 255.255.255.252
  ip route 10.20.0.0 255.255.0.0 10.1.1.6

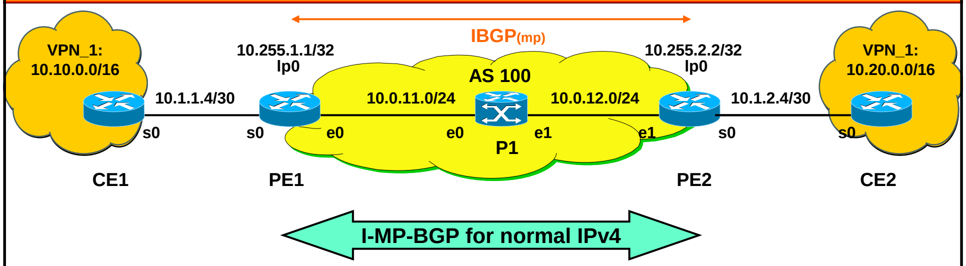
PE1 (OSPF and MPLS in Backbone):
ip cef
int loopback 0
  ip address 10.255.1.1 255.255.255.255
int s0
  ip address 10.1.1.6 255.255.255.252
int e0
  ip address 10.0.11.1 255.255.255.0
  mpls ip
router ospf 100
  network 10.0.11.1 0.0.0.0 area 0

CE2:
int s0
  ip address 10.1.2.5 255.255.255.252
  ip route 10.10.0.0 255.255.0.0 10.1.2.6

PE2 (OSPF and MPLS in Backbone):
ip cef
int loopback 0
  ip address 10.255.2.2 255.255.255.255
int s0
  ip address 10.1.2.6 255.255.255.252
int e0
  ip address 10.0.12.2 255.255.255.0
  mpls ip
router ospf 100
  network 10.0.12.1 0.0.0.0 area 0

© 2006, D.I. Manfred Lindner MPLS_VPN, v4.6 81

Start Normal I-BGP in AS 100



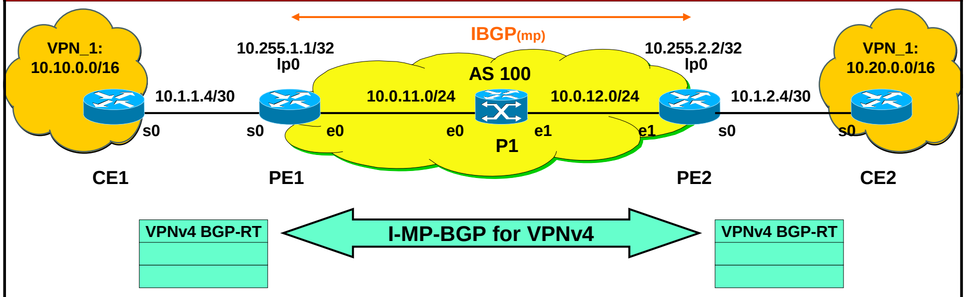
PE1:
int loopback 0
  ip address 10.255.1.1 255.255.255.255

router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
  address-family ipv4
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family

PE2:
int loopback 0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family ipv4
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family

© 2006, D.I. Manfred Lindner MPLS_VPN, v4.6 82

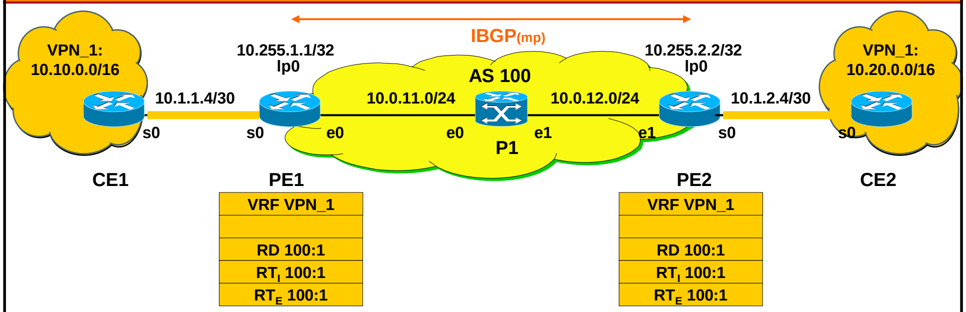
Start MP-BGP in AS 100



PE1:
```
int loopback 0
  ip address 10.255.1.1 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
 address-family vpnv4
    neighbor 10.255.2.2 activate
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 send-community extended (default)
 exit-address-family
```

PE2:
```
int loopback 0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
 address-family vpnv4
    neighbor 10.255.1.1 activate
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 send-community extended
 exit-address-family
```


Create VRF and Bring Interface into VRF (PE router)

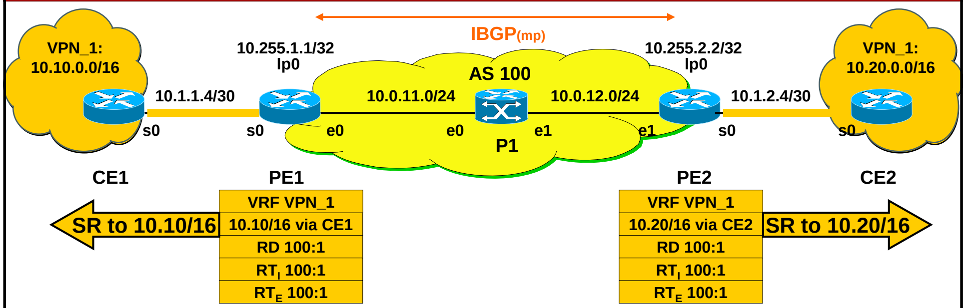


PE1:
```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.1.6 255.255.255.252
```

PE2:
```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.2.6 255.255.255.252
```

L86 - MPLS VPN

Static Routing (SR) towards CE (PE router)



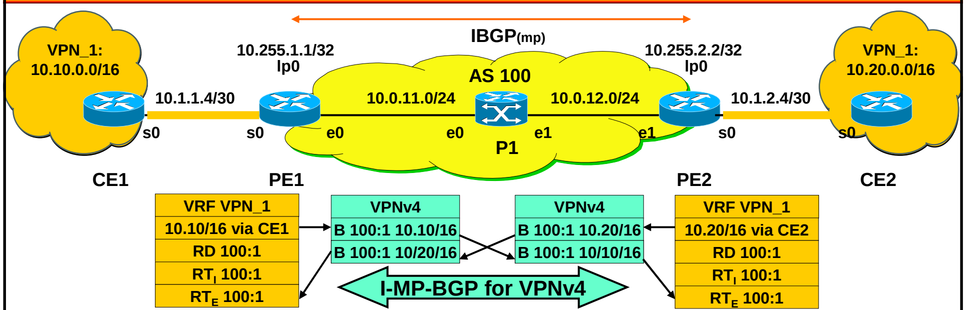
PE1:

```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.1.6 255.255.255.252

ip route vrf VPN_1 10.10.0.0 255.255.0.0 serial 0 10.1.1.5
```

PE2:

```
ip vrf VPN_1
  rd 100:1
  route-target import 100:1
  route-target export 100:1

int s0
  ip vrf forwarding VPN_1
  ip address 10.1.2.6 255.255.255.252

ip route vrf VPN_1 10.20.0.0 255.255.0.0 serial 0 10.1.2.5
```


Redistributing Static into MP-BGP and Transport of Static routes via I-MP-BGP (PE router)



PE1:

```
ip vrf VPN_1
  rd 100:1

router bgp 100
  address-family ipv4 vrf VPN_1
    redistribute static
    no auto-summary
    no synchronization
  exit-address-family
```

PE2:

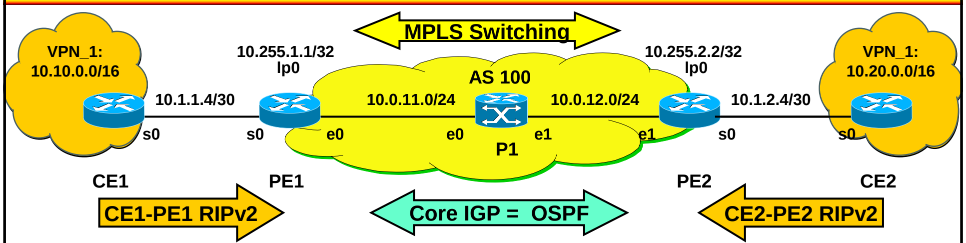
```
ip vrf VPN_1
  rd 100:1

router bgp 100
  address-family ipv4 vrf VPN_1
    redistribute static
    no auto-summary
    no synchronization
  exit-address-family
```

Agenda

- MP-BGP
- VPN Overview
- MPLS VPN Architecture
- MPLS VPN Basic VPNs
- MPLS VPN Complex VPNs
- MPLS VPN Configuration (Cisco)
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

 MPLS_VPN, v4.6

IP Addressing, RIPv2 Routing in VPN_1, Basic OSPF Routing and MPLS in AS 100



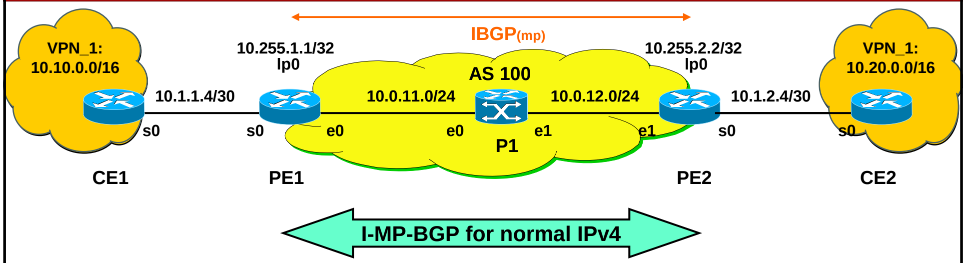

```
CE1:
int s0
  ip address 10.1.1.5 255.255.255.252
router rip
  version 2
  network 10.0.0.0
  no auto-summary

PE1:
ip cef
int loopback 0
  ip address 10.255.1.1 255.255.255.255
int s0
  ip address 10.1.1.6 255.255.255.252
int e0
  ip address 10.0.11.1 255.255.255.0
  mpls ip
router ospf 100
  network 10.0.11.1 0.0.0.0 area 0
```

```
CE2:
int s0
  ip address 10.1.2.5 255.255.255.252
router rip
  version 2
  network 10.0.0.0
  no auto-summary

PE2:
ip cef
int loopback 0
  ip address 10.255.2.2 255.255.255.255
int s0
  ip address 10.1.2.6 255.255.255.252
int e0
  ip address 10.0.12.2 255.255.255.0
  mpls ip
router ospf 100
  network 10.0.12.1 0.0.0.0 area 0
```

 MPLS_VPN, v4.6

L86 - MPLS VPN

Start Normal I-BGP in AS 100



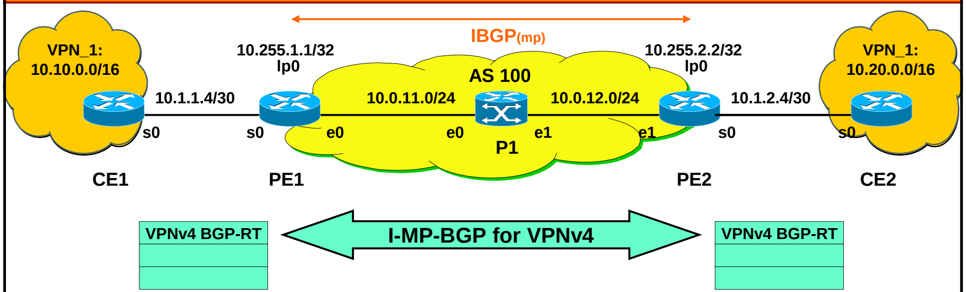
PE1:
int loopback 0
  ip address 10.255.1.1 255.255.255.255

router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
  address-family ipv4
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family

PE2:
int loopback 0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family ipv4
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family

© 2006, D.I. Manfred Lindner MPLS_VPN, v4.6 89

Start MP-BGP in AS 100

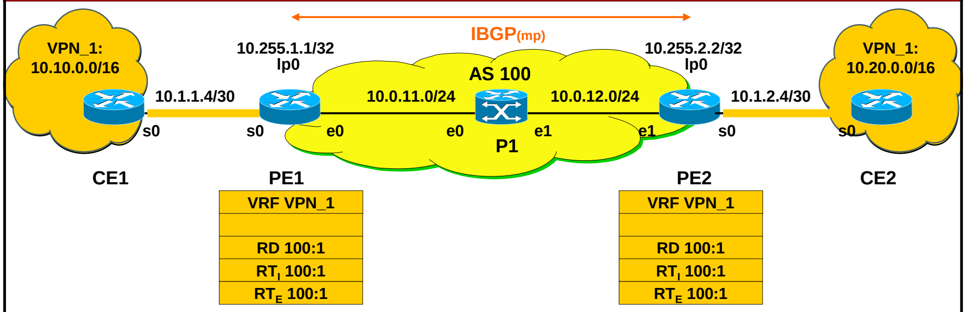


PE1:
int loopback 0
  ip address 10.255.1.1 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
  address-family vpnv4
    neighbor 10.255.2.2 activate
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 send-community extended (default)
  exit-address-family

PE2:
int loopback 0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family vpnv4
    neighbor 10.255.1.1 activate
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 send-community extended
  exit-address-family

© 2006, D.I. Manfred Lindner MPLS_VPN, v4.6 90

Create VRF and Bring Interface into VRF (PE router)



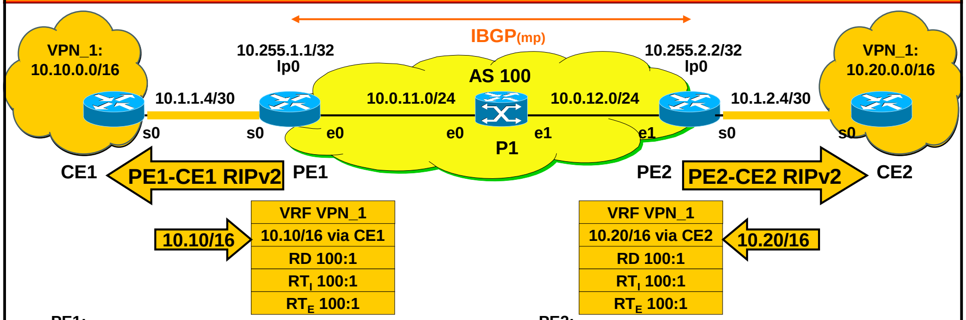
PE1:

```
ip vrf VPN_1
 rd 100:1
 route-target import 100:1
 route-target export 100:1

int s0
 ip vrf forwarding VPN_1
 ip address 10.1.1.6 255.255.255.252
```

PE2:

```
ip vrf VPN_1
 rd 100:1
 route-target import 100:1
 route-target export 100:1

int s0
 ip vrf forwarding VPN_1
 ip address 10.1.2.6 255.255.255.252
```


Start Dynamic Routing (RIPv2) towards CE (PE router)



PE1:

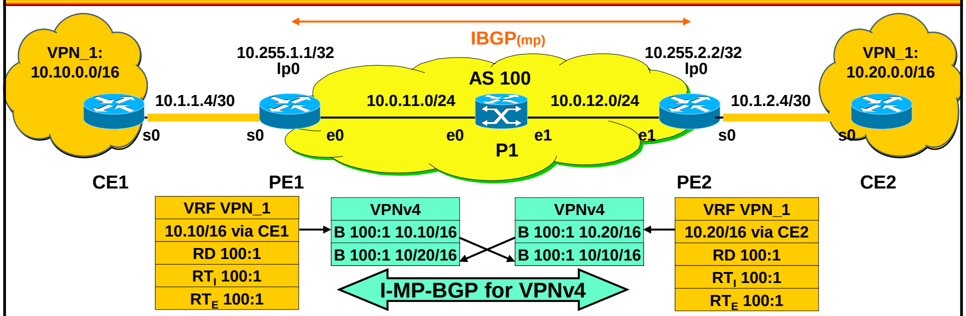
```
ip vrf VPN_1
 rd 100:1

int s0
 ip vrf forwarding VPN_1
 ip address 10.1.1.6 255.255.255.252

router rip
 version 2
 address-family ipv4 vrf VPN_1
   network 10.0.0.0
   no auto-summary
 exit-address-family
```

PE2:

```
ip vrf VPN_1
 rd 100:1

int s0
 ip vrf forwarding VPN_1
 ip address 10.1.2.6 255.255.255.252

router rip
 version 2
 address-family ipv4 vrf VPN_1
   network 10.0.0.0
   no auto-summary
 exit-address-family
```

L86 - MPLS VPN

Redistributing VRF RIPv2 into MP-BGP and Transport of VPNv4 routes via I-MP-BGP (PE router)



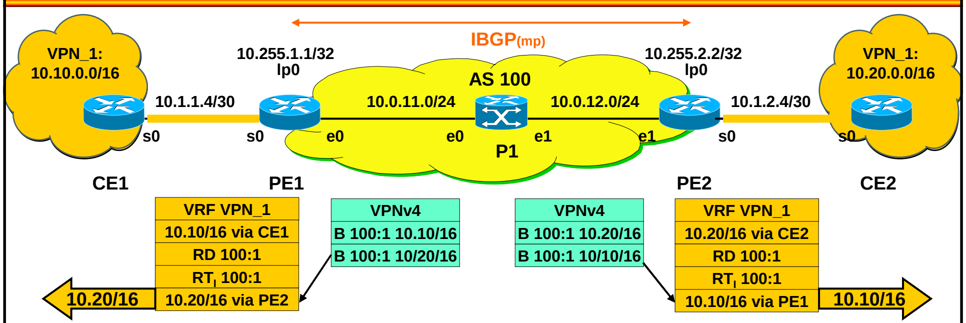
PE1:
ip vrf *VPN_1*
rd 100:1

```
router bgp 100
 no bgp default ipv4-unicast
 address-family ipv4 vrf VPN_1
  redistribute rip
  no auto-summary
  no synchronization
 exit-address-family
```

PE2:
ip vrf *VPN_1*
rd 100:1

```
router bgp 100
 no bgp default ipv4-unicast
 address-family ipv4 vrf VPN_1
  redistribute rip
  no auto-summary
  no synchronization
 exit-address-family
```


Redistribution of VPNv4 routes into VRF RIPv2 (PE router)



PE1:

```
router rip
 version 2
 address-family ipv4 vrf VPN_1
  redistribute bgp 100 metric transparent
 exit-address-family

router bgp 100
 no bgp default ipv4-unicast
 address-family ipv4 vrf VPN_1
  redistribute rip
 exit-address-family
```

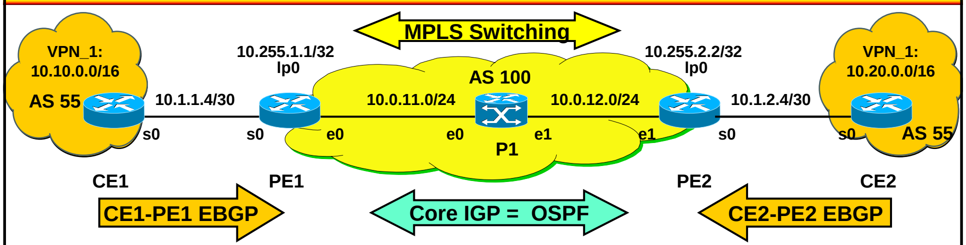
PE2:

```
router rip
 version 2
 address-family ipv4 vrf VPN_1
  redistribute bgp 100 metric transparent
 exit-address-family

router bgp 100
 no bgp default ipv4-unicast
 address-family ipv4 vrf VPN_1
  redistribute rip
 exit-address-family
```

Agenda

- MP-BGP
- VPN Overview
- MPLS VPN Architecture
- MPLS VPN Basic VPNs
- MPLS VPN Complex VPNs
- MPLS VPN Configuration (Cisco)
  - CE-PE OSPF Routing
  - CE-PE Static Routing
  - CE-PE RIP Routing
  - CE-PE External BGP Routing

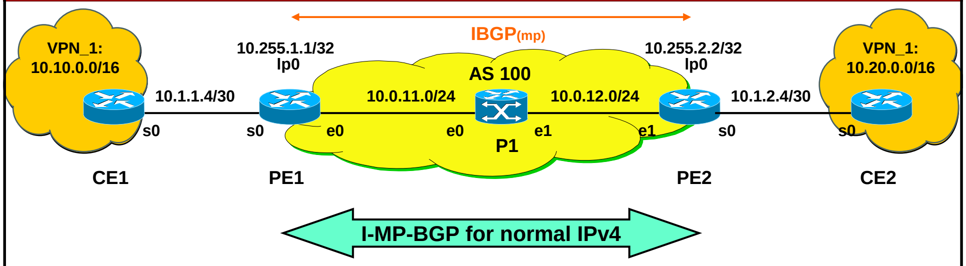
IP Addressing, EBGP Routing in VPN_1, Basic OSPF Routing and MPLS in AS 100




```
CE1:
int s0
  ip address 10.1.1.5 255.255.255.252
router bgp 55
  neighbor 10.1.1.6 remote-as 100
  network 10.10.0.0 mask 255.255.0.0

PE1:
ip cef
int loopback 0
  ip address 10.255.1.1 255.255.255.255
int s0
  ip address 10.1.1.6 255.255.255.252
int e0
  ip address 10.0.11.1 255.255.255.0
  mpls ip
router ospf 100
 network 10.0.11.1 0.0.0.0 area 0
```

```
CE2:
int s0
  ip address 10.1.2.5 255.255.255.252
router bgp 55
  neighbor 10.1.2.6 remote-as 100
  network 10.20.0.0 mask 255.255.0.0

PE2:
ip cef
int loopback 0
  ip address 10.255.2.2 255.255.255.255
int s0
  ip address 10.1.2.6 255.255.255.252
int e0
  ip address 10.0.12.2 255.255.255.0
  mpls ip
router ospf 100
 network 10.0.12.1 0.0.0.0 area 0
```

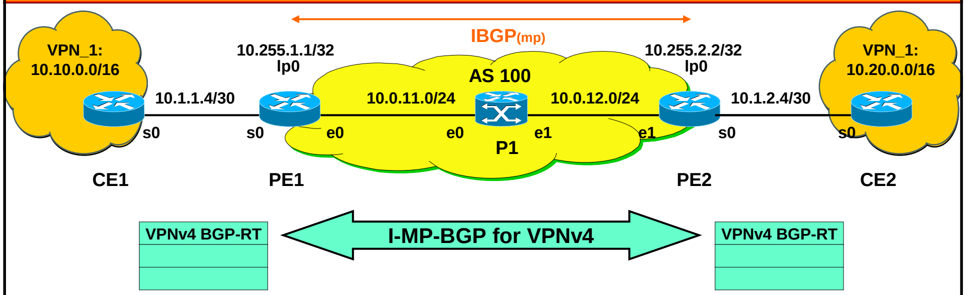
Start Normal I-BGP in AS 100




```
PE1:
int loopback 0
  ip address 10.255.1.1 255.255.255.255

router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
  address-family ipv4
    neighbor 10.255.2.2 next-hop-self
    neighbor 10.255.2.2 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family
```

```
PE2:
int loopback 0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family ipv4
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 activate
    no auto-summary (default)
    no synchronization (default)
  exit address-family
```

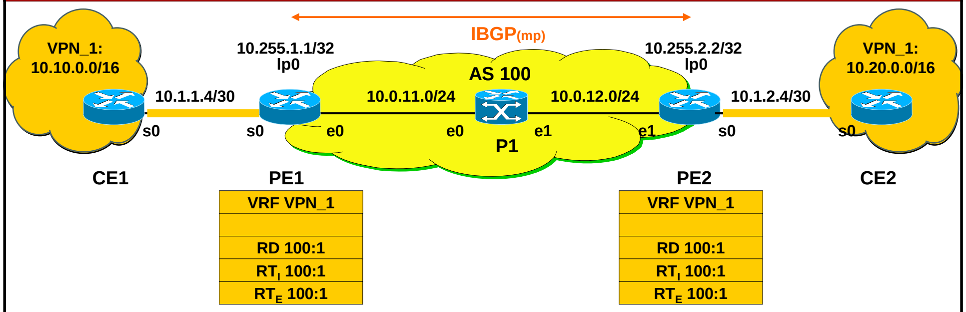

Start MP-BGP in AS 100




```
PE1:
int loopback 0
  ip address 10.255.1.1 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.1.1
  neighbor 10.255.2.2 remote-as 100
  neighbor 10.255.2.2 update-source loop 0
 address-family vpnv4
   neighbor 10.255.2.2 activate
   neighbor 10.255.2.2 next-hop-self
   neighbor 10.255.2.2 send-community extended (default)
 exit-address-family
```

```
PE2:
int loopback 0
  ip address 10.255.2.2 255.255.255.255
router bgp 100
  no bgp default ipv4-unicast
  bgp router-id 10.255.2.2
  neighbor 10.255.1.1 remote-as 100
  neighbor 10.255.1.1 update-source loop 0
  address-family vpnv4
    neighbor 10.255.1.1 activate
    neighbor 10.255.1.1 next-hop-self
    neighbor 10.255.1.1 send-community extended
  exit-address-family
```

Create VRF and Bring Interface into VRF (PE router)



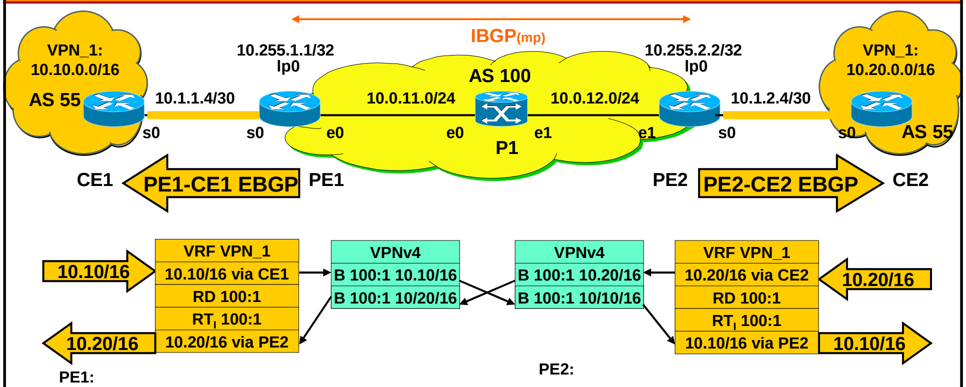
PE1:

```
ip vrf VPN_1
 rd 100:1
 route-target import 100:1
 route-target export 100:1

int s0
 ip vrf forwarding VPN_1
 ip address 10.1.1.6 255.255.255.252
```

PE2:

```
ip vrf VPN_1
 rd 100:1
 route-target import 100:1
 route-target export 100:1

int s0
 ip vrf forwarding VPN_1
 ip address 10.1.2.6 255.255.255.252
```


Start Dynamic Routing towards CE using EBGP Redistribute into MP-BGP and vice versa



PE1:

```
ip vrf VPN_1
 rd 100:1

router bgp 100
 address-family ipv4 vrf VPN_1
  neighbor 10.1.1.5 remote-as 55
  neighbor 10.1.1.5 activate
  no auto-summary
  no synchronization
exit-address-family
```

PE2:

```
ip vrf VPN_1
 rd 100:1

router bgp 100
 address-family ipv4 vrf VPN_1
  neighbor 10.1.2.5 remote-as 55
  neighbor 10.1.2.5 activate
  no auto-summary
  no synchronization
exit-address-family
```