Peer to Peer VPNs

Enterprise Networks (6CFU)

Slides taken from Prof. Andrea Detti’s TPI2 lectures.

Introduction

- **Allow secure traffic exchange among company branches ditributed over the entire territory**
- **Usually required by business customers**
- **Virtual Private Networks**
  - » **Private: allows communication between subnets in different networks as they were in the same private network (as for addressing, routing and security)**
  - » **Virtual: the required links between networks are (necessarily) virutal (not phisical). The support network is not private.**

- **Private IP addressing**
  - » **10.0.0.0/8**
  - » **172.16.0.0/12**
  - » **192.168.0.0/16**

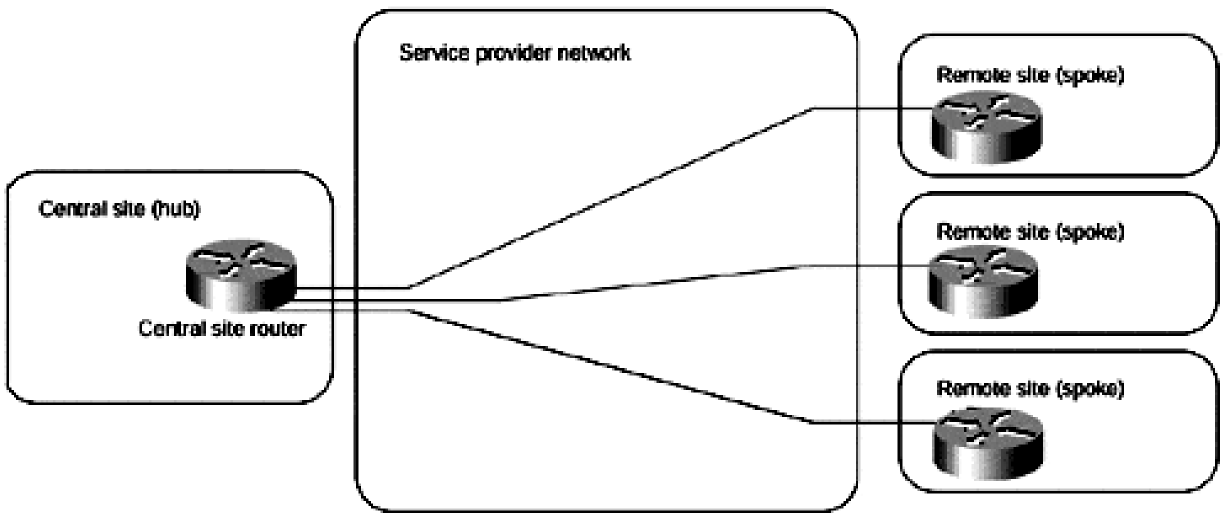
- **Requirements: unique addressing in a VPN**

VPN models

- **Communication models**
  - » **Intra-company (Intranet)**
  - » **Inter-company (Extranet)**
    - » **Addresses must be unique**
  - » **VPDN (Virtual Private Dialup Network)**
    - » **Dynamic address configuration**
- **Data transfer models**
  - » **Overlay: ISP network is used only for transporting features. Routing information is exchanged between company networks. The VPN topology composed by point to point links configured by the customers**
  - » **Peer-to-peer: the ISP is responsible also for exchangin routing information Logical topology is defined by the customers. Physical topology is defined by the ISP**

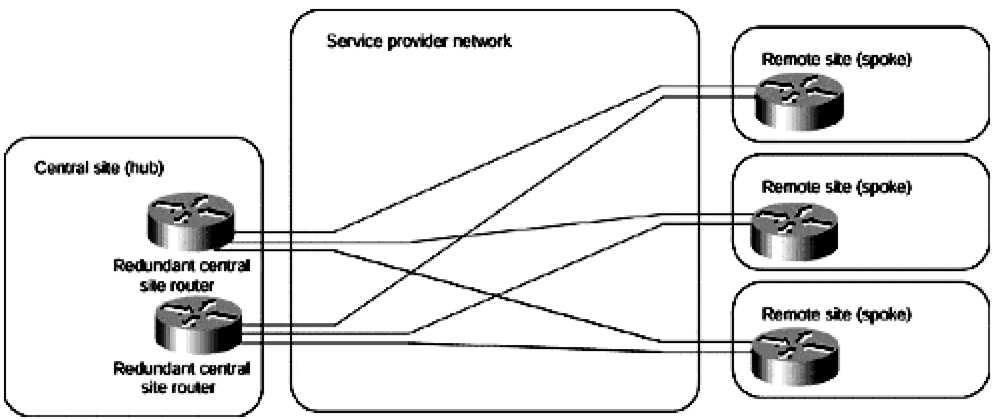
VPN topologies – Hub and Spoke

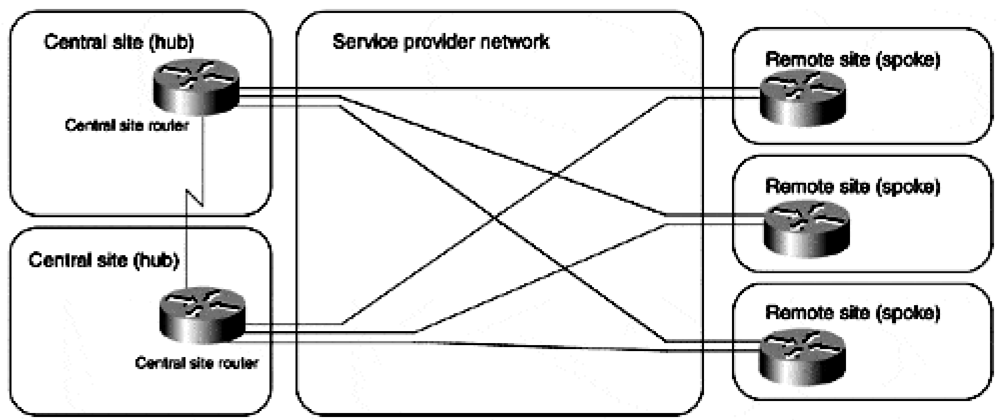
- VPN topology depends on the specific customers' needs. Nevertheless, there are some "standard" topologies…
- Hub-and-spoke Topology:
  - » Remote branches (spoke) connected to a central site (hub).
  - » Spokes can communicate with each other, but inter-spoke should be negligible then spoke-hub traffic



VPN topologies – Hub and Spoke

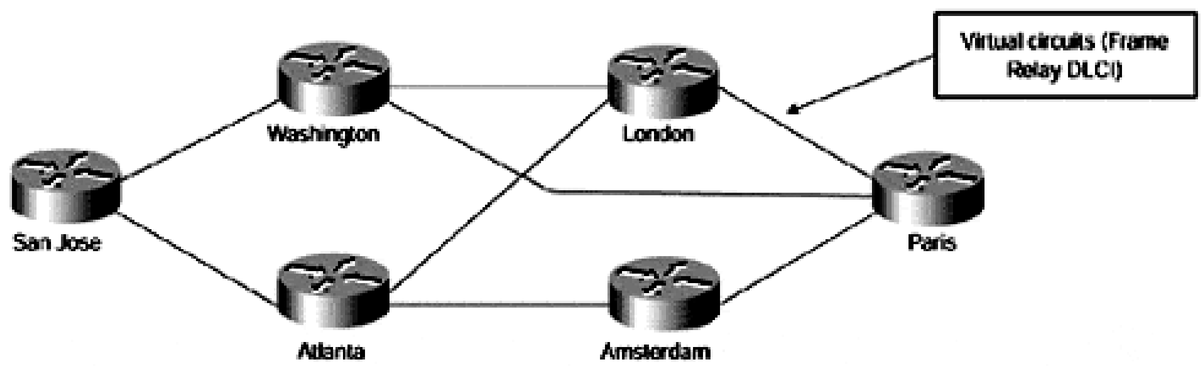
- **Hub Backup**





VPN topologies – Partial/Full-Mesh

- When there's a huge data exchange between enterprises' sites, the Hub-and-Spoke topology is less effective since all the spoke-to-spoke traffic traverses the hub → bottleneck
- In such a case, partially or totally connected topologies are preferred
- Business case:
  - » Companies without a strict gerarchic organization
  - » Peer-to-peer applications (messaging or collaboration system)
  - » For multinational companies in which the cost of the hub-and-spoke solution could be high because of the excessive cost of international links.

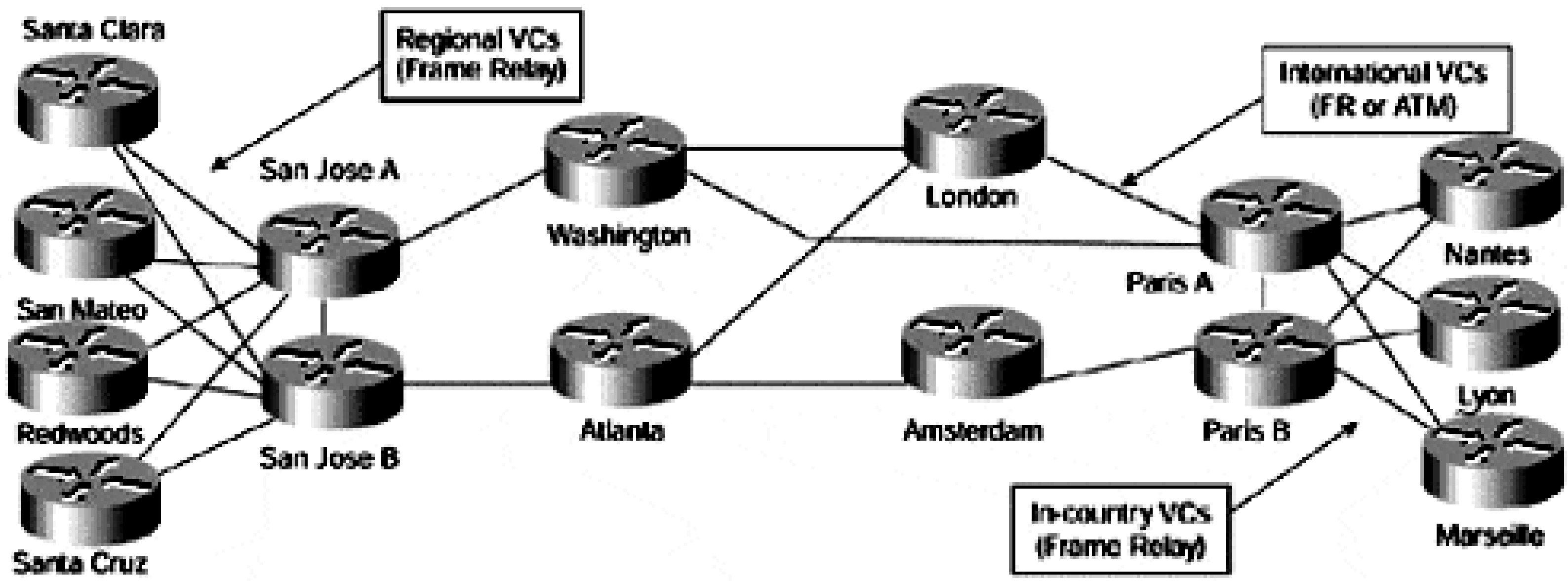


VPN topologies – Partial/Full-Mesh

- The full-mesh topology is simple to plan
  - » Get the traffic matrix $A(i,j)=x$ Mb and ask to the ISP a link between site *i* and site *j* with *x* Mbps
- BUT... the full-mesh cost can be high since the number of links employed is n*(n-1)
- SO… often a partial mesh topology is adopted
- How to plan a partial mesh topology?
  - » 1) Create a fully connected topology through links only between sites having a huge traffic exchange
  - » 2) From the traffic matrix, and assuming a shortest-path routing, compute the amount of required bandwidth on all the employed links
  - » 3) Order links from the cheapest ISP ;-)

VPN topologies – Hybrid

- Huge international VPNs are often composed by many national hub-and-spoke VPNs
- The international part (backbone) is a partial-mesh between hubs



Peer-to-Peer VPN

Peer-to-Peer VPN

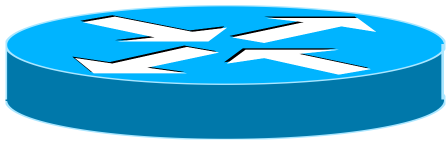
- Routing Information exchange between Company and ISP routers → routing happens on a layer composed both by company entities and by ISP entities

- De facto based on BGP/MPLS solution
  - » Enterprise's gateway transfers data to the ISP which handles the forwarding through other Enterprise's sites
  - » Routing (connections topology) is actually in the hands of the ISP
  - » Plug & Play, adding a site is a matter of ISP configuration only, the company has to do almost nothing

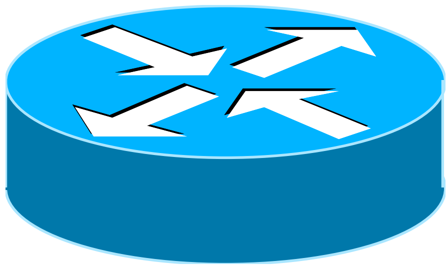
VPN BGP/MPLS

Peer-to-Peer VPN

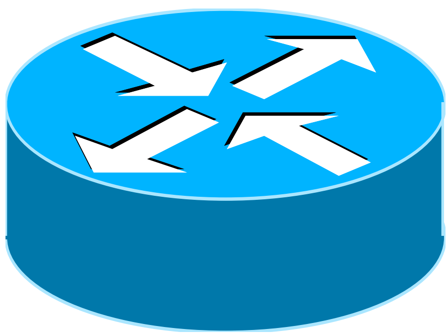
Elements of a VPN BGP/MPLS network



Customer Edge : is the Company side router facing with the ISP which provides the VPN BGP/MPLS service. It has standard routing functionalities; its only peer is the Provider edge with which exchanges info through BGP messages



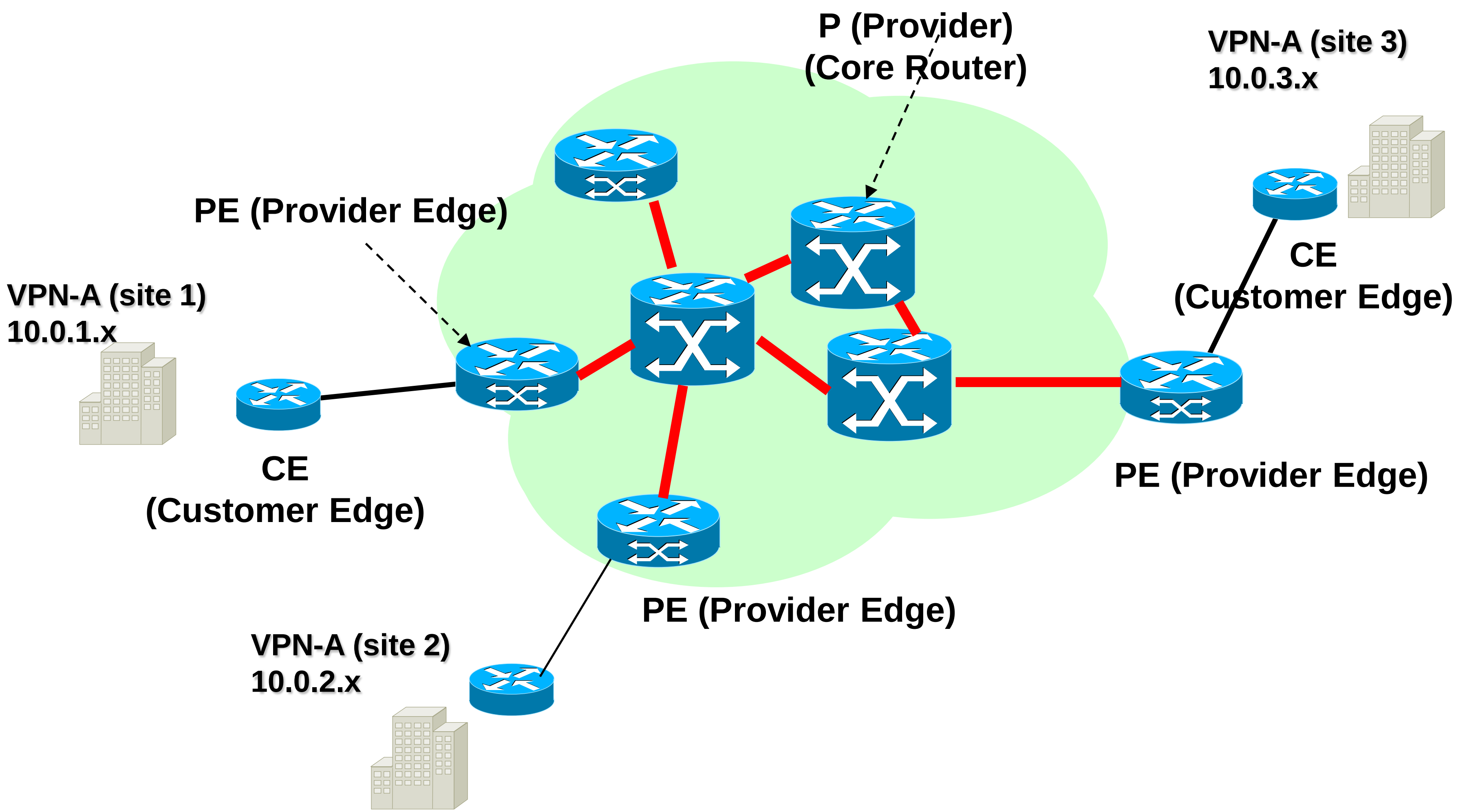
Provider Edge : is the access router on the ISP side in which one or more Customer Edges are connected. Besides IP functionalities, it also handles the MPLS LER role.



Provider Router : Label Switched Router (LSR) composing the MPLS backbone of the ISP

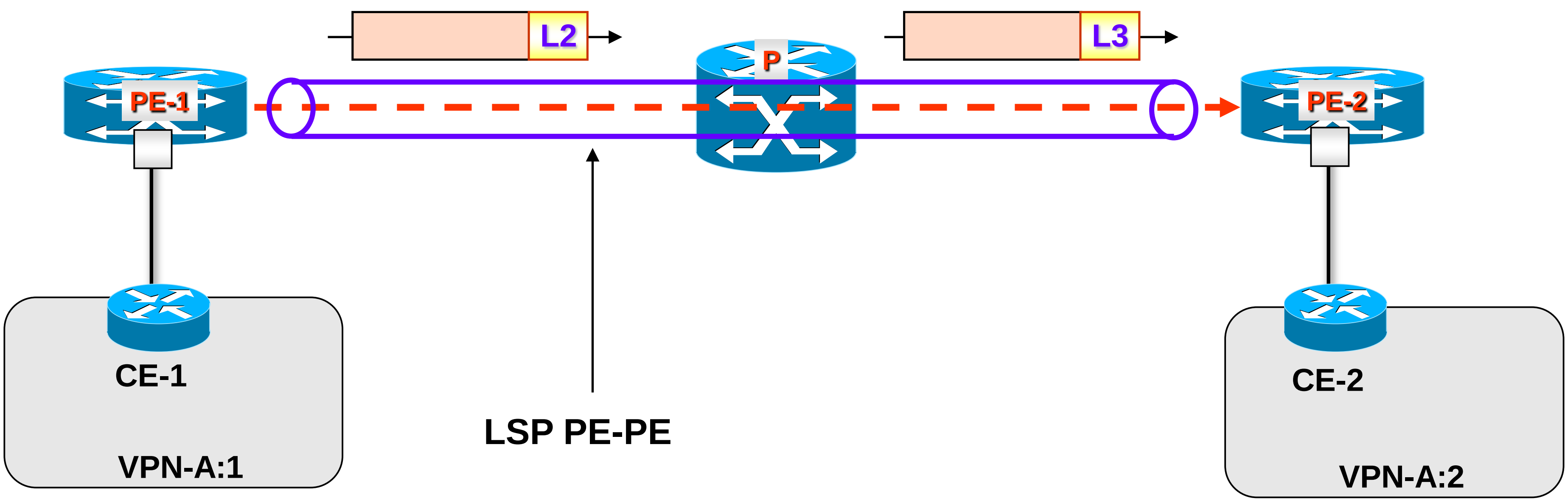
MPLS/VPN Backbone : MPLS network with properly configures LSPs to interconnect all the Provider Edges.

VPN MPLS service architecture

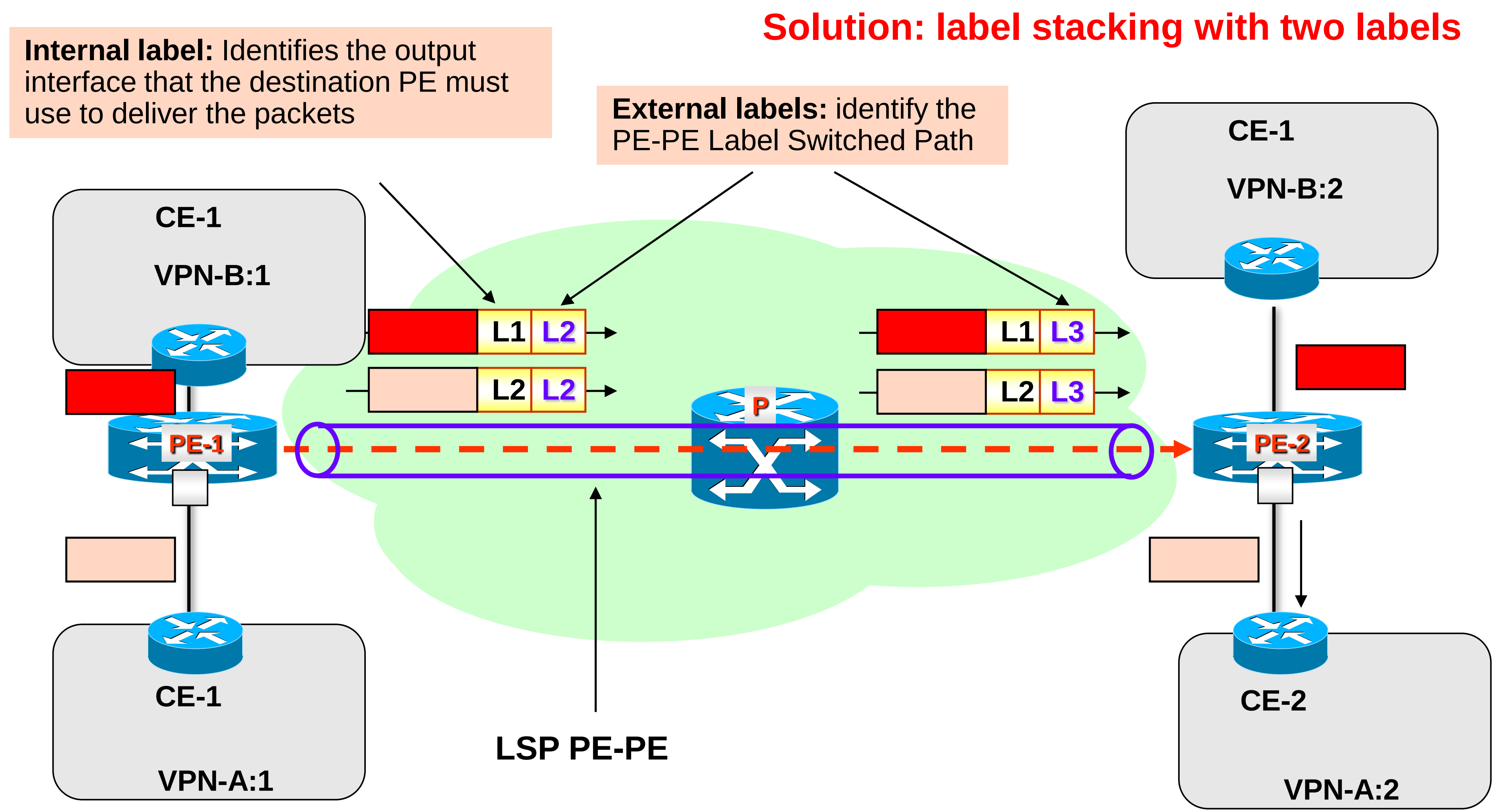


Forwarding mechanism

- **Problem: transfer packets between the two sites of a VPN → *A:site1 --- A:site2***
- **Trivial solution (A:1 → A:2): encapsulate at PE (A:1) IP packets coming from CE(A:1) in the ISP, linking PE(A:1)→PE(A:2)**
- **At the LSP end, PE(A:2) forwards on an IP base**
- **What happens if same PEs support more than a VPN with *non-coordinated* addressing?**
- **It can happen that PE(A:2) finds itself to forward (on IP base) packets with the same network addesses, BUT different VPNs!!!**



Forwarding mechanism

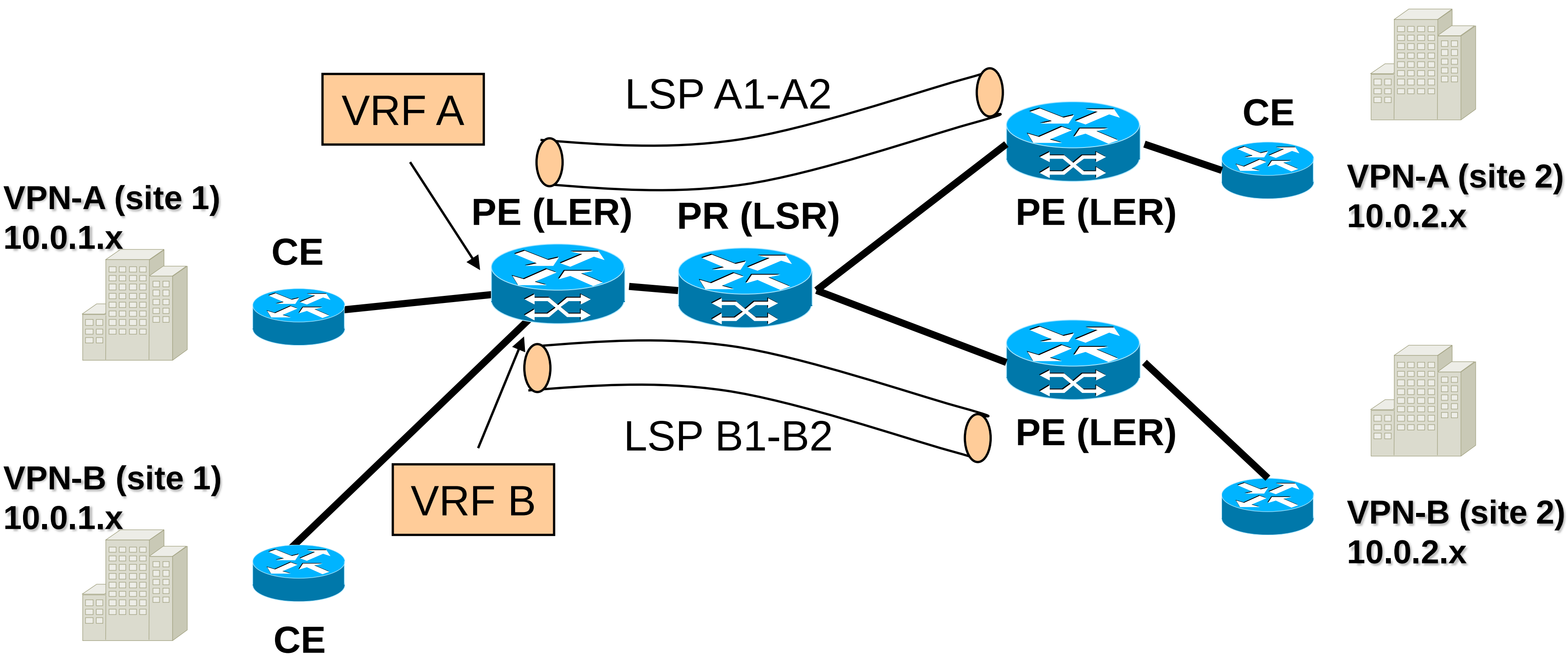


PE classification

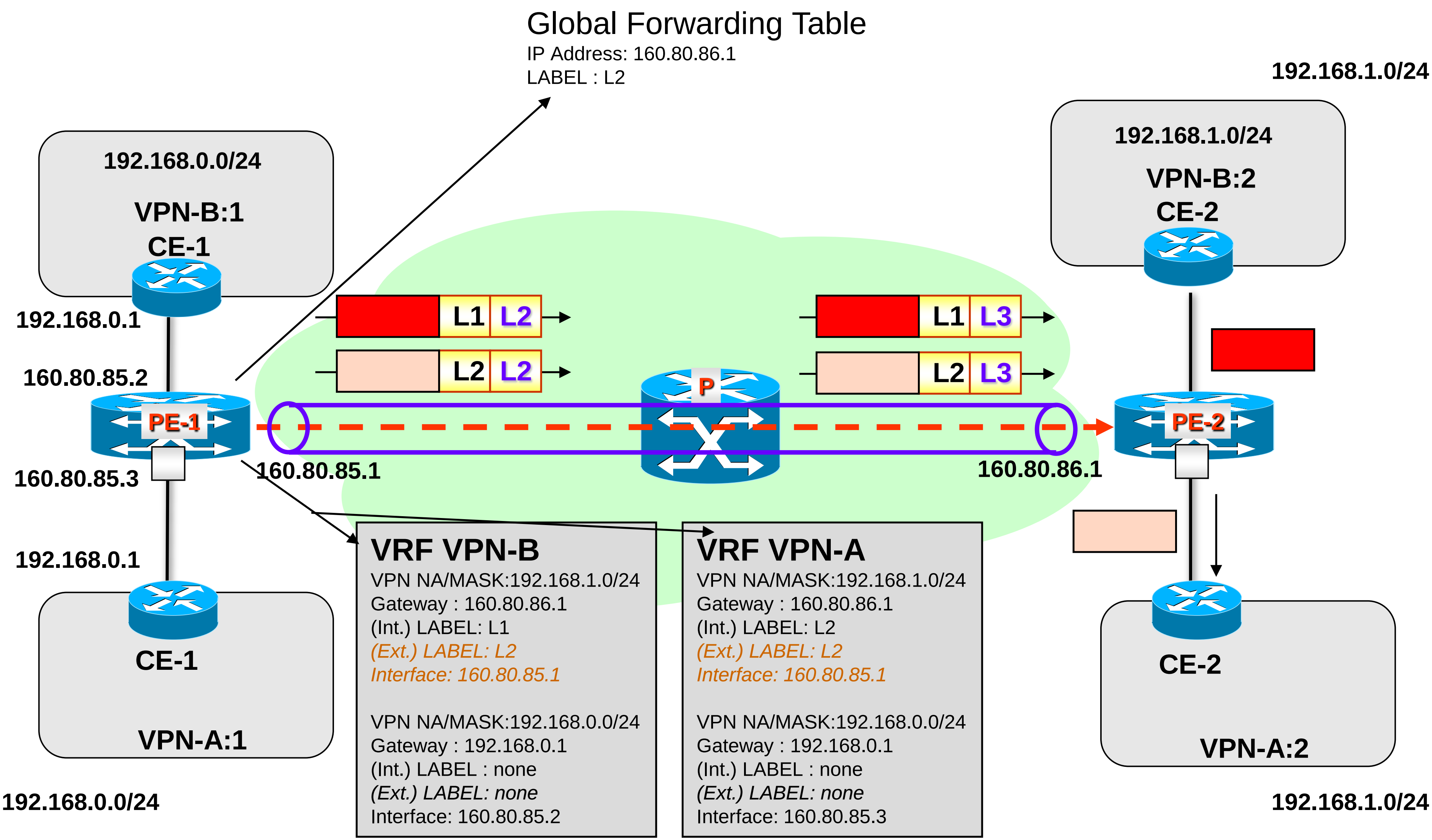
- **PROBLEM: how can the PE forward/classify the packets coming from CE A:1 on the right tunnel?**
- **SOLUTION: it must know to which VPN packets belong to**
  - » **this information is deduced from the ingress interface in which a packet is received**

- **SO, depending on the belonging VPN, the MPLS forwarding of the packet changes. Technically, the PE stores as many forwarding tables as the number of VPNs connected to it. Each *virtual* table is named VPN Routing and Forwarding (VRF) table**
  - » **A VRF entry contains (logically) the following tuple: <VPN network address, VPN mask, Next PE IP Address, Internal label, Output Interface>**
- **In addition to the VRF, a PE stores one Global Forwarding Table (GRT) which permits to reach a PE from another PE.**
  - » **Logically, a GRT entry contains the tuple: <PE IP address, external label, Output Interface>**

PE classification



VRF and GFT

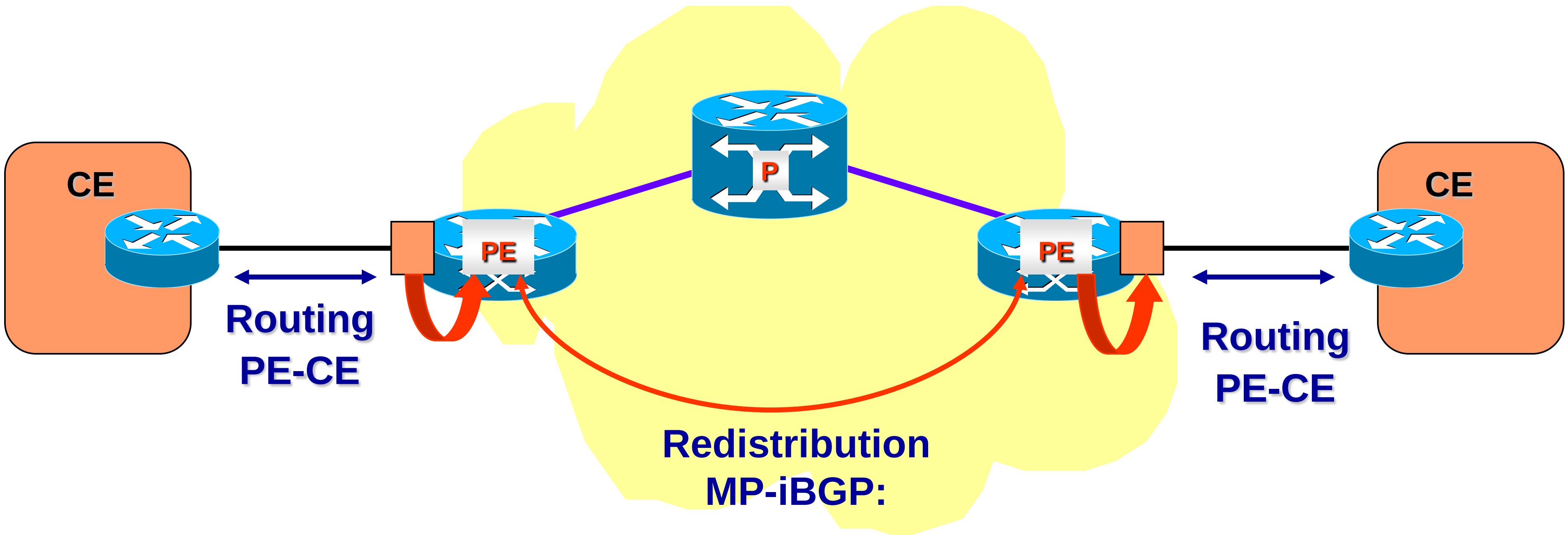


Global Forwarding Table
IP Address: 160.80.86.1
LABEL : L2
192.168.1.0/24
192.168.0.0/24
VPN-B:1
CE-1
192.168.1.0/24
VPN-B:2
CE-2
L1 L2
L2 L2
L1 L3
L2 L3
192.168.0.1
160.80.85.2
PE-1
P
PE-2
160.80.85.3
160.80.85.1
160.80.86.1
192.168.0.1
CE-1
VPN-A:1
CE-2
VPN-A:2
192.168.0.0/24
192.168.1.0/24
VRF VPN-B
VPN NA/MASK:192.168.1.0/24
Gateway : 160.80.86.1
(Int.) LABEL: L1
(Ext.) LABEL: L2
Interface: 160.80.85.1
VPN NA/MASK:192.168.0.0/24
Gateway : 192.168.0.1
(Int.) LABEL : none
(Ext.) LABEL: none
Interface: 160.80.85.2
VRF VPN-A
VPN NA/MASK:192.168.1.0/24
Gateway : 160.80.86.1
(Int.) LABEL: L2
(Ext.) LABEL: L2
Interface: 160.80.85.1
VPN NA/MASK:192.168.0.0/24
Gateway : 192.168.0.1
(Int.) LABEL : none
(Ext.) LABEL: none
Interface: 160.80.85.3

Populating GFT and VRFs

- The Global Forwarding Table is configured by the provider during the set-up o the MPLS/VPN backbone (i.e. LSPs between PEs)
- The GFT can be populated manually (in the case of manual LSPs), or automatically in the case of a set-up with signalling protocols like LDP, RSVP-TE or CR-LDP
- VRFs contain two forwarding categories:
  - » Forwading to LOCAL sites
  - » Forwarding to REMOTE sites
- Forwarding to local sites can be:
  - » Manually configured
  - » Obtained through specific routing protocols (OSPF, RIP, etc.), running the CE-PE link
- Remote routes are obtained through an extension of the BGP-4 protocol, namely Multi-Protocol interior BGP (MP-iBGP or MP-BGP)

Populating VRFs



- Routing CE-PE: Static, RIP, OSPF, eBGP
- Routing PE-PE: MP-iBGP = MultiProtocol-internal BGP

Populating VRFs

- **VRFs "synchronize" among them exchanging the reachability info inside MP-iBGP announces**

- **An MP-iBGP announce is sent by a PE to all other PEs; i.e. it exists an overlay full mesh between PEs**

- **Assumption: the cost of the *direct hop* between two PEs is 1, being this an IP level hop (not MPLS hop)**

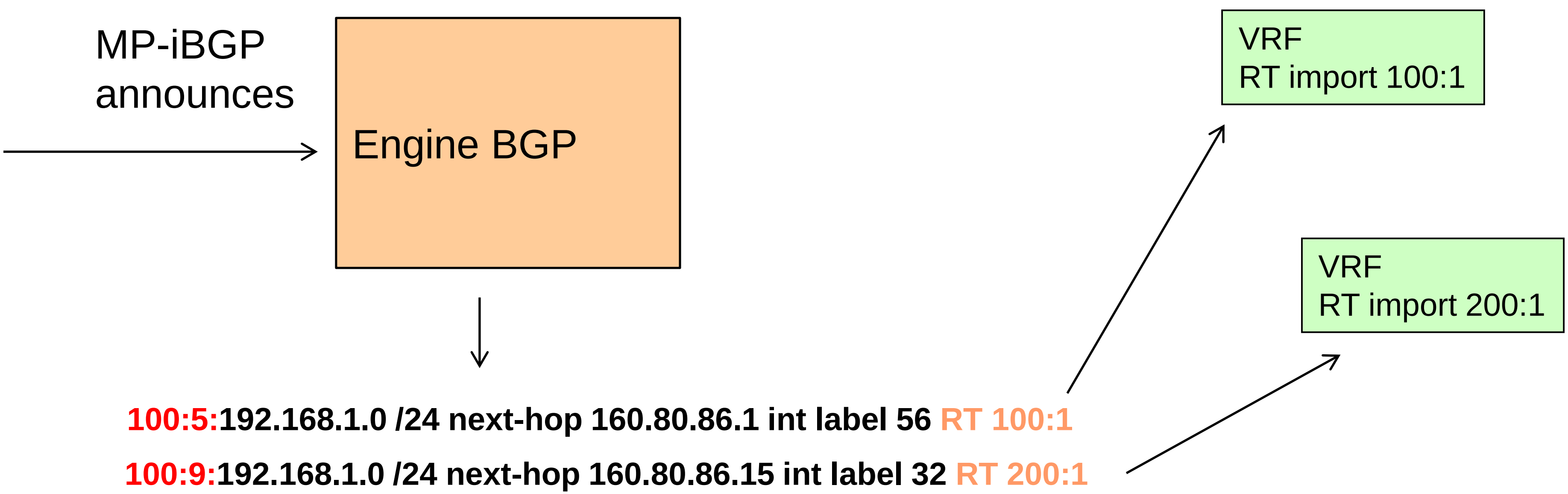
- **A same MP-iBGP announce carries reachability information relative to prefixes of more VRFs**

Route Distinguisher

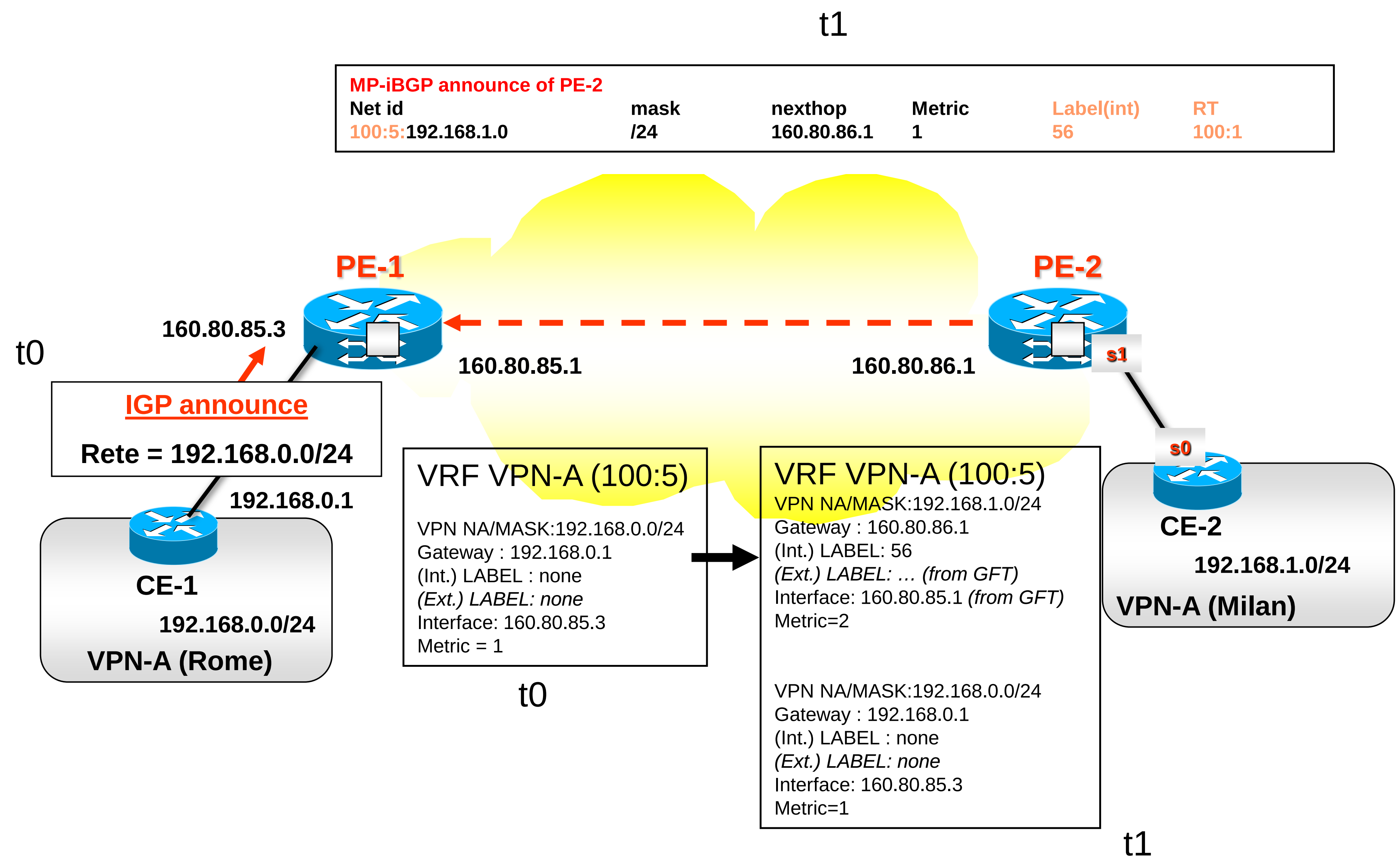
- **Thanks to MP-iBGP announces, the BGP engine inside the PE calculates the next-hop (and internal label) towards every announced prefix**
- **VRFs belonging to different VPNs can notify a same private prefix since the addressing spaces can be overlapped.**
- **To differentiate overlapped prefixes (i.e. make them different to the BGP engine), a VRF is characterized by an ID named Route Distinguisher (64 bit)**
  - » **Usually, all the VRFs of the *same* VPN use the *same* Route Distinguisher, since the prefixes inside a VPN cannot overlap.**
  - » **In this way, the Route Distinguisher can be reused**

Route Distinguisher

- **The RD is placed before the *net_id* in the MP-iBGP entries**
- **The routes computed by BGP are inserted inside the enabled VRFs (see Route Target next…)**



Populating VRFs



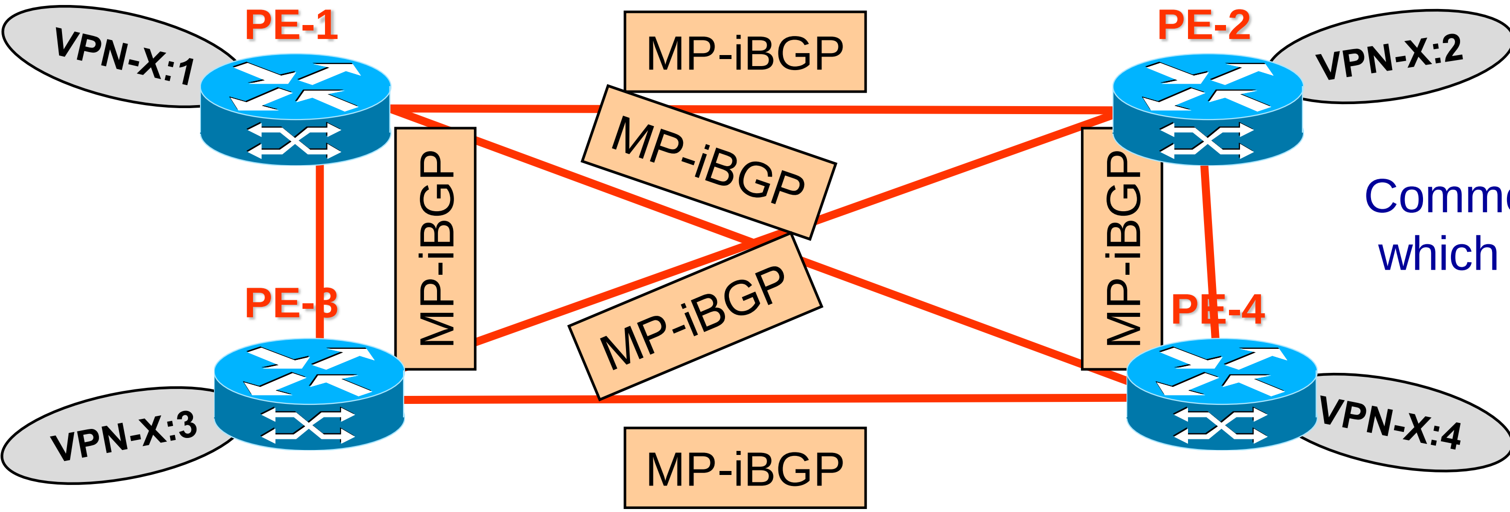
What about the VPN topology?

- **If MP-iBGP messages are diffused among all PEs, all the VPNs have a full-mesh topology**
- **PROBLEM: what if I want different topologies for different VPNs?**
- **BGP priciples say that if I have an overlay topology in which MP-iBGP messages are diffused, the (forwarding) topology of VPN-x is the set of the *overlay* shortest-paths between any couple of nodes**
- **Since direct connections between two PEs have metric 1 → the VPN-x topology matches the overlay topology in which MP-iBGP messages are notified**
- **Therefore, if the overlay network in which MP-iBGP messages are forwarded is full-mesh, the VPN topology is full-mesh, too**

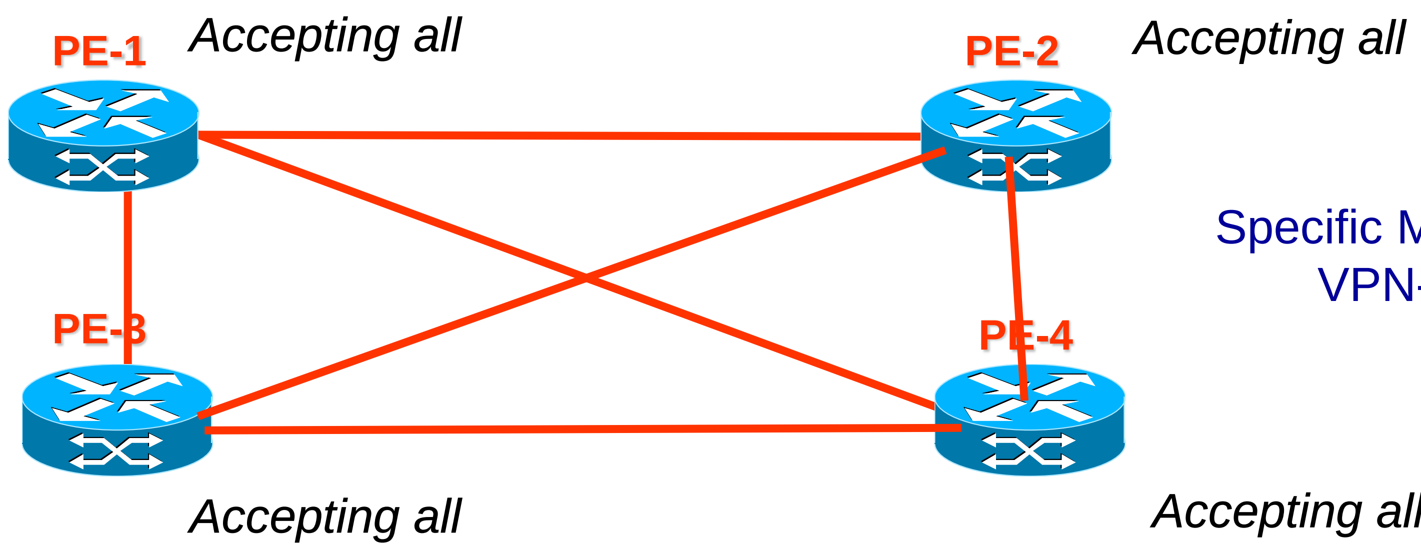
What about the VPN topology?

- **To change the logical topology of VPN-x it is necessary to change the MP-iBGP overlay network of VPN-x**
  - » **Solution 1: create a different MP-iBGP overlay forwarding topology for each VPN**
    - » **Cons: high management effort, cannot aggregate inside the same MP-iBGP message the routing information relative to more VPNs, etc.**
  - » **Solution 2:**
    - » **Having an overlay full-mesh for MP-iBGP *common* between PEs**
    - » **Define the *specific* overlay needed for a given VPN**
    - » **Flood MP-iBGP messages on the *common* MP-iBGP overlay**
    - » **Receivers elaborate only announces coming from links of the *specific* overlay**

Populating VRFs - VPN Full Mesh



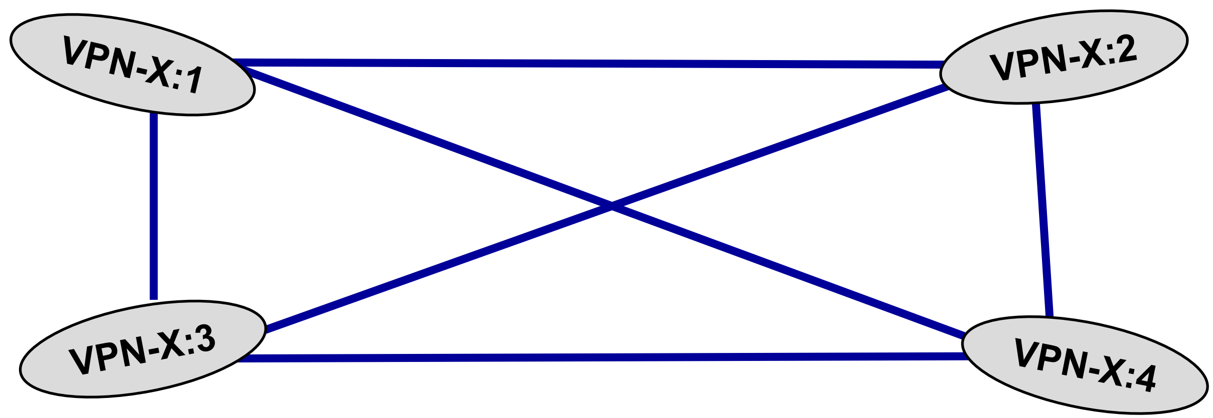
Common MP-iBGP Overlay in which messages are flooded



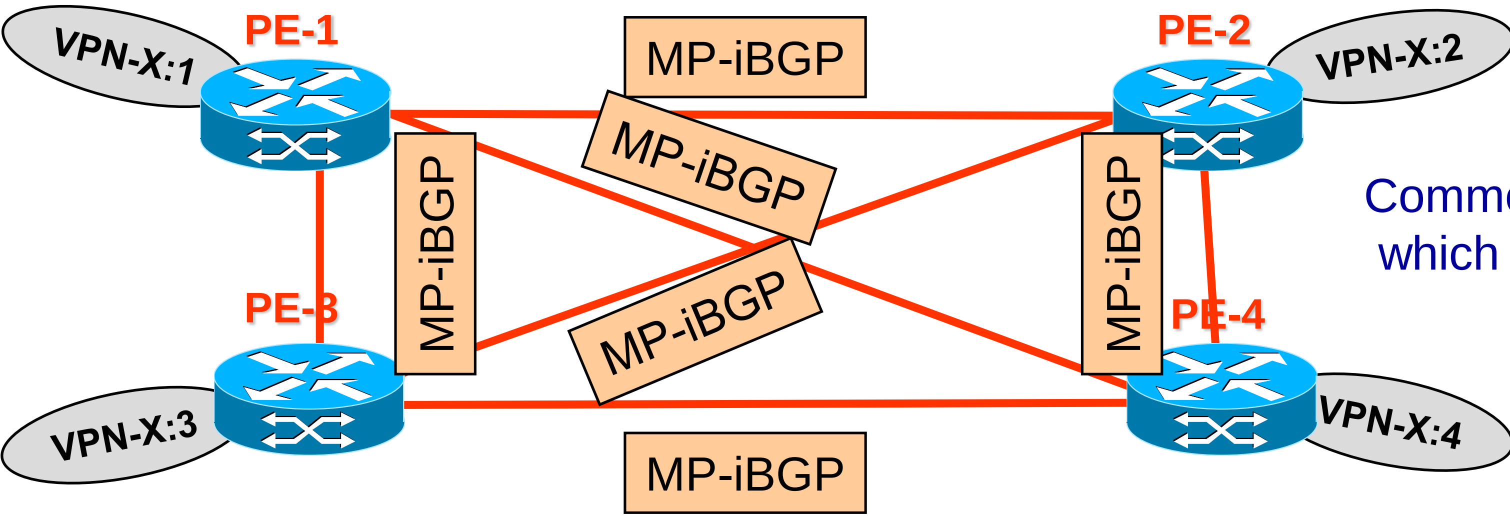
Specific MP-iBGP overlay of VPN-x (full MESH)

Populating VRFs - VPN Full Mesh

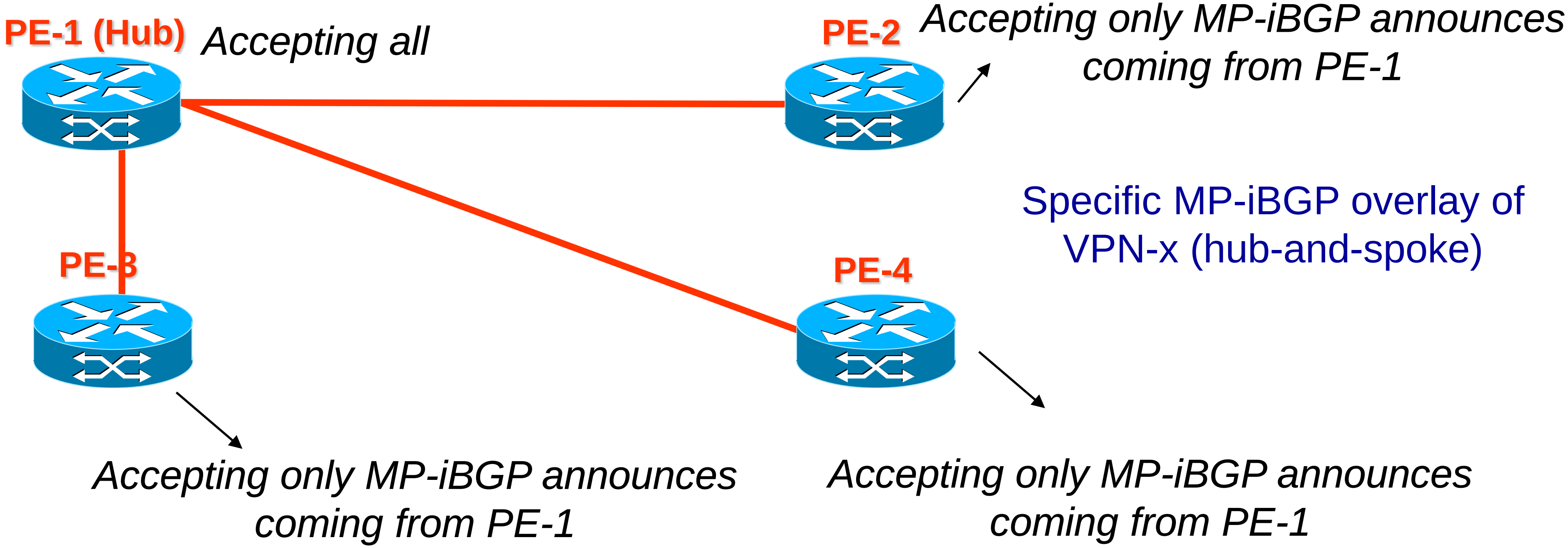
Resulting VPN topology



Populating VRFs - VPN Hub and Spoke



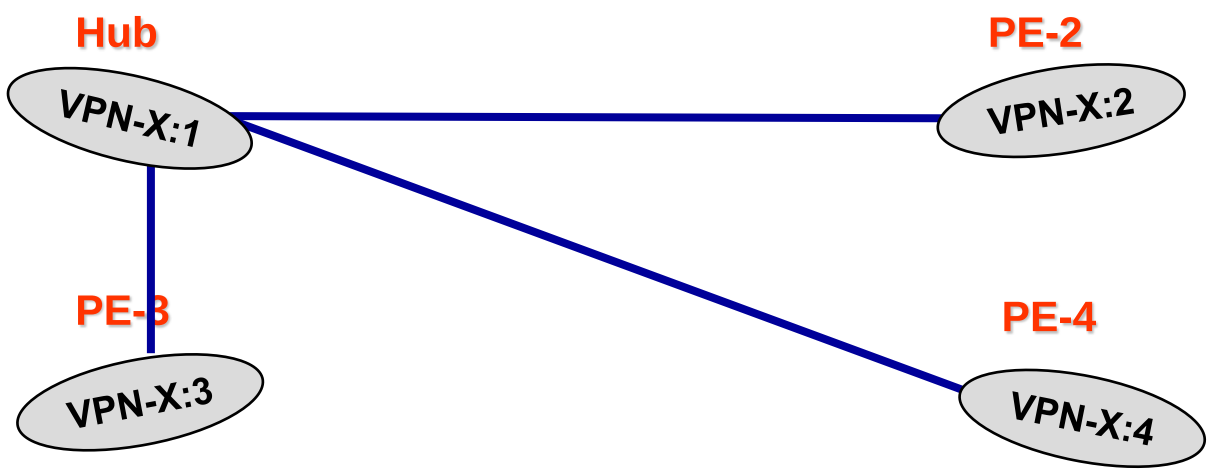
Common MP-iBGP Overlay in which messages are flooded



Specific MP-iBGP overlay of VPN-x (hub-and-spoke)

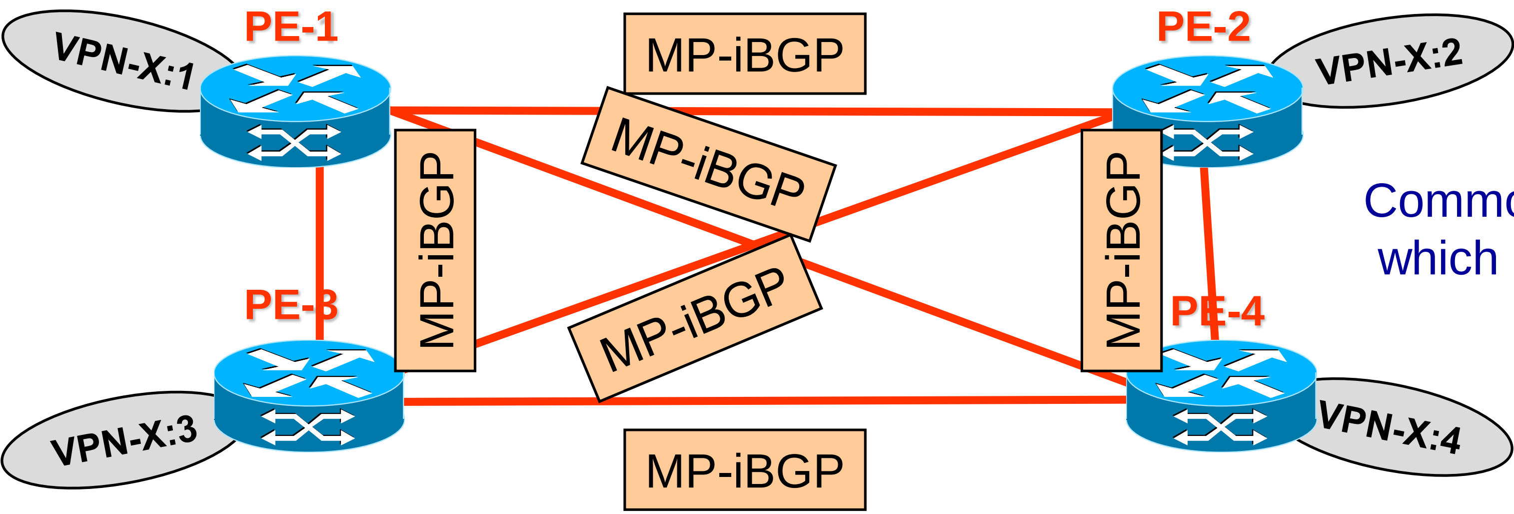
Populating VRFs - VPN Hub and Spoke

Resulting VPN topology

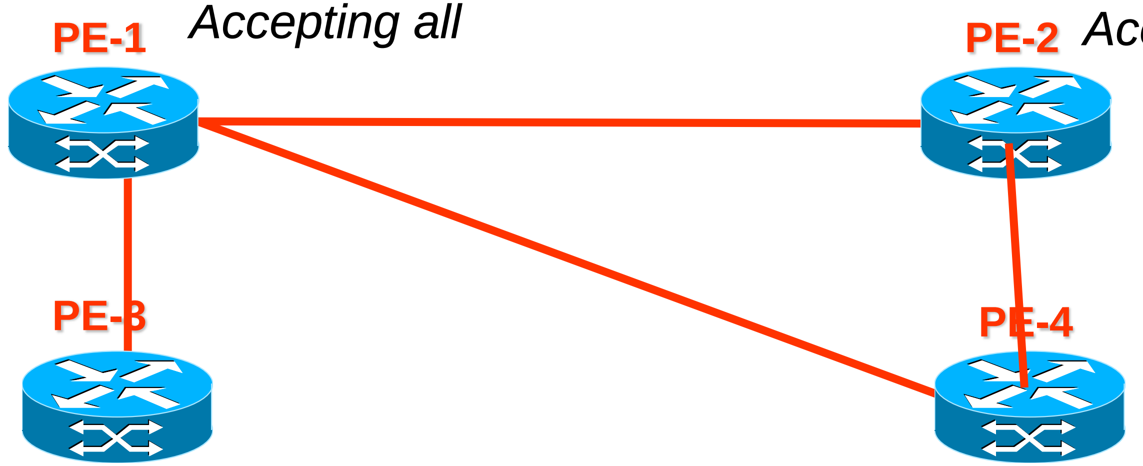


NOTE: MP-iBGP announces (like iBGP) cannot propagate on iBGP links. Therefore, in order to permit to spokes the communication between them, The hub's VRF must export a default route

Populating VRFs - VPN Partial Mesh



Common MP-iBGP Overlay in which messages are flooded



Accepting all

Accepting only MP-iBGP announces coming from PE-1 and PE-4

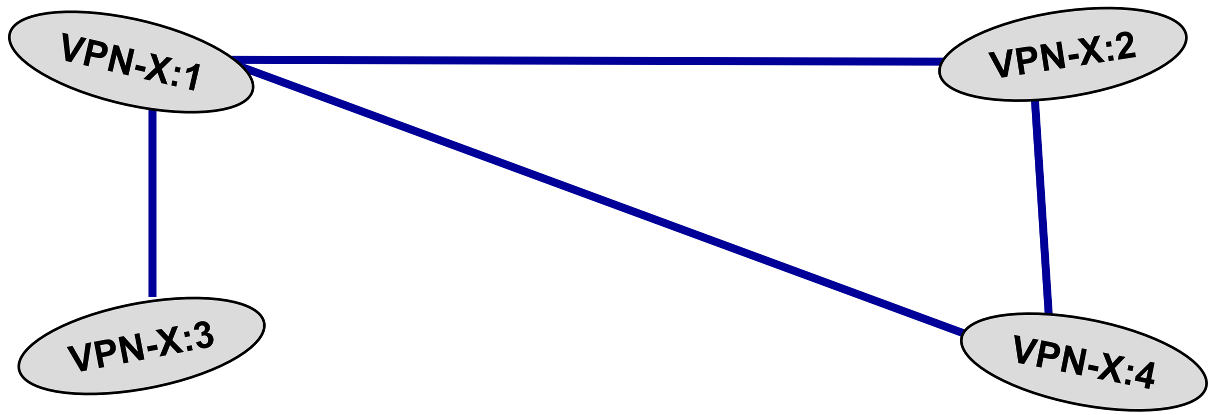
Specific MP-iBGP overlay of VPN-x (Partial MESH)

Accepting only MP-iBGP announces coming from PE-1

Accepting only MP-iBGP announces coming from PE-1 and PE-2

Populating VRFs - VPN Partial Mesh

Resulting VPN topology

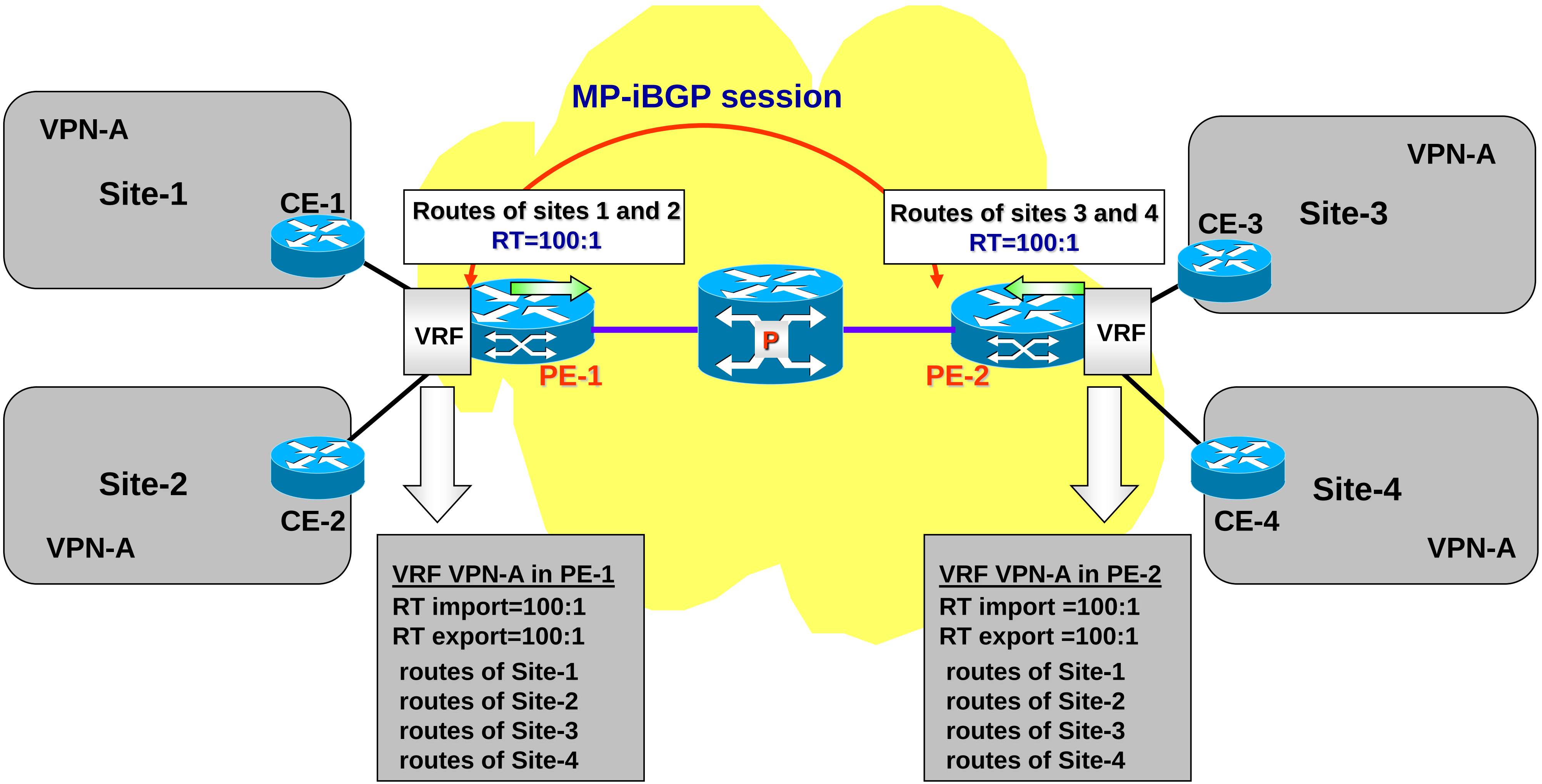


Route Target

- <u>The Route Target concept</u> permits to realize a specific overlay for the VPN-x discussed before. Therefore, <u>permits to define VPN-x topology.</u>
- It’s the VPN/MPLS “way” to tell to a VRF-x to “accept only a subset of MP-iBGP announces”
- HOW:
  - » Each VRF transmitting announces, labels (exports) these announces with a configurable ID (Route target) of 64 bit size
  - » Each VRF can receive (import) only announces with a configurable subset of Route Targets

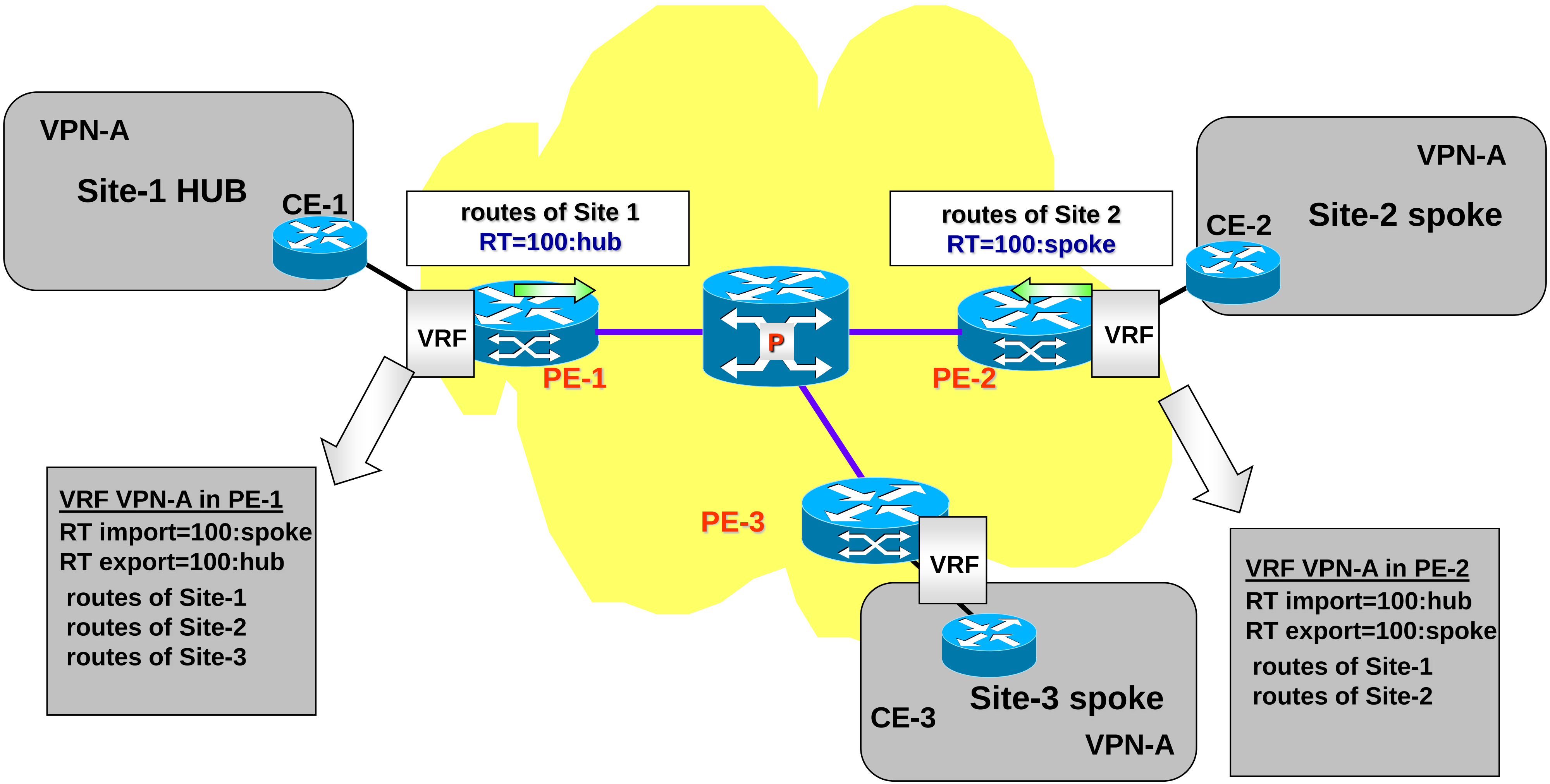
Using the “Route Target”: Example 1

- VPN full mesh



Using the “Route Target”: Example 2

- VPN Hub and spoke



VPN/MPLS configuration

- **Initialization**
  - » **Configure LSP MPLS (e.g. with LDP) between all PEs**
  - » **Enable BGP peering for prefixes of type *VPNv4* (RD+net_id) between all PEs**
- **How to add another site:**

Client

- » **Notify to ISP the need of another VPN site and the relative topology**
- » **Install a CE as enterprise gateway**
- » **Configure the *default gateway* of the CE with the IP address of the access PE**
- » ***Optional:* enable on CE a routing protocol on the CE-PE path (e.g. OSPF)**

Provider

- » **Initialize a new VRF on access PE**
- » **Define/Configure the Route Distinguisher**
- » **Define/Configure Route Import and Route Export and eventually update the import/export RTs on the other PEs, coherently with the requested topology**
- » **Associate the ingress PE interface with the VRF**
- » **Enable MP-iBGP on such VRF**

VPN/MPLS conclusions

- Simple approach for the client
- Security of traffic is delegated to the ISP (not to be handled by the client!)
- Modest configuration complexity by the provider
- Provider must «traffic engineer» the VPN/MPLS backbone so to offer the promised QoS to the VPN clients
  - » Traffic engineering in the LSPs between Pes
- The cost could be high, though

Cisco IOS configuration

On all involved PE

- **Create user VRF**
  - » PE-1(config)# ip vrf vpnB
  - » PE-1(config-vrf)#rd 200:0
  - » PE-1(config-vrf)#route-target import 200:2
  - » PE-1(config-vrf)#route-target export 200:1
  - » PE-1(config-vrf)#exit
- **Add to the VRF a manual route towars local CE in case of no routing protool on the PE-CE link**
  - » PE-1(config)#ip route vrf vpnB 192.168.0.0 255.255.255.0 160.2.11.2
- **Associate interface to the VRF**
  - » PE-1(config)#int f0/1
  - » PE-1(config-if)#ip vrf forwarding vpnB
  - » PE-1(config-if)#ip address 160.2.11.1 255.255.255.25

Cisco IOS configuration

- Configure BGP
- Optionally disable IPv4 peering
  - router bgp 3269
    - no bgp default ipv4-unicast
- Create peering with all PEs (if not existent)
  - neighbor 2.2.2.2 remote-as 3269
  - neighbor 2.2.2.2 update-source Loopback0
  - neighbor 3.3.3.3 remote-as 3269
  - neighbor 3.3.3.3 update-source
- Activate vpnv4 peerings
  - address-family vpnv4
    - neighbor 2.2.2.2 activate
    - neighbor 2.2.2.2 send-community extended
    - neighbor 2.2.2.2 next-hop-self
    - exit-address-family

Cisco IOS configuration

- **Switch on the BGP advertisements of VRF in case eBGP was not active in the PE-CE link**
  - » **address-family ipv4 vrf vpnB**
  - » **network 192.168.0.0**

- **In case BGP was active on PE-CE**
  - » **Configure BGP global peering with PE**
    - » **neighbor 160.2.11.2 remote-as 200**
    - » **neighbor 160.2.11.2 update-source FastEthernet0/1**
  - » **Bind VRF to the neighbour**
    - » **address-family ipv4 vrf vpnB**
    - » **neighbor 160.2.11.2 remote-as 200**
    - » **neighbor 160.2.11.2 activate**
    - » **neighbor 160.2.11.2 as-override**

Cisco IOS configuration

- **Debug**
  - » **show ip vrf**
  - » **show ip route vrf vpnB**
  - » **show mpls forwarding-table**
    - » **Useful to know the external label towards the remote PE**
  - » **show ip bgp vpnv4 vrf vpnB labels**
    - » **Useful to know the internal label**