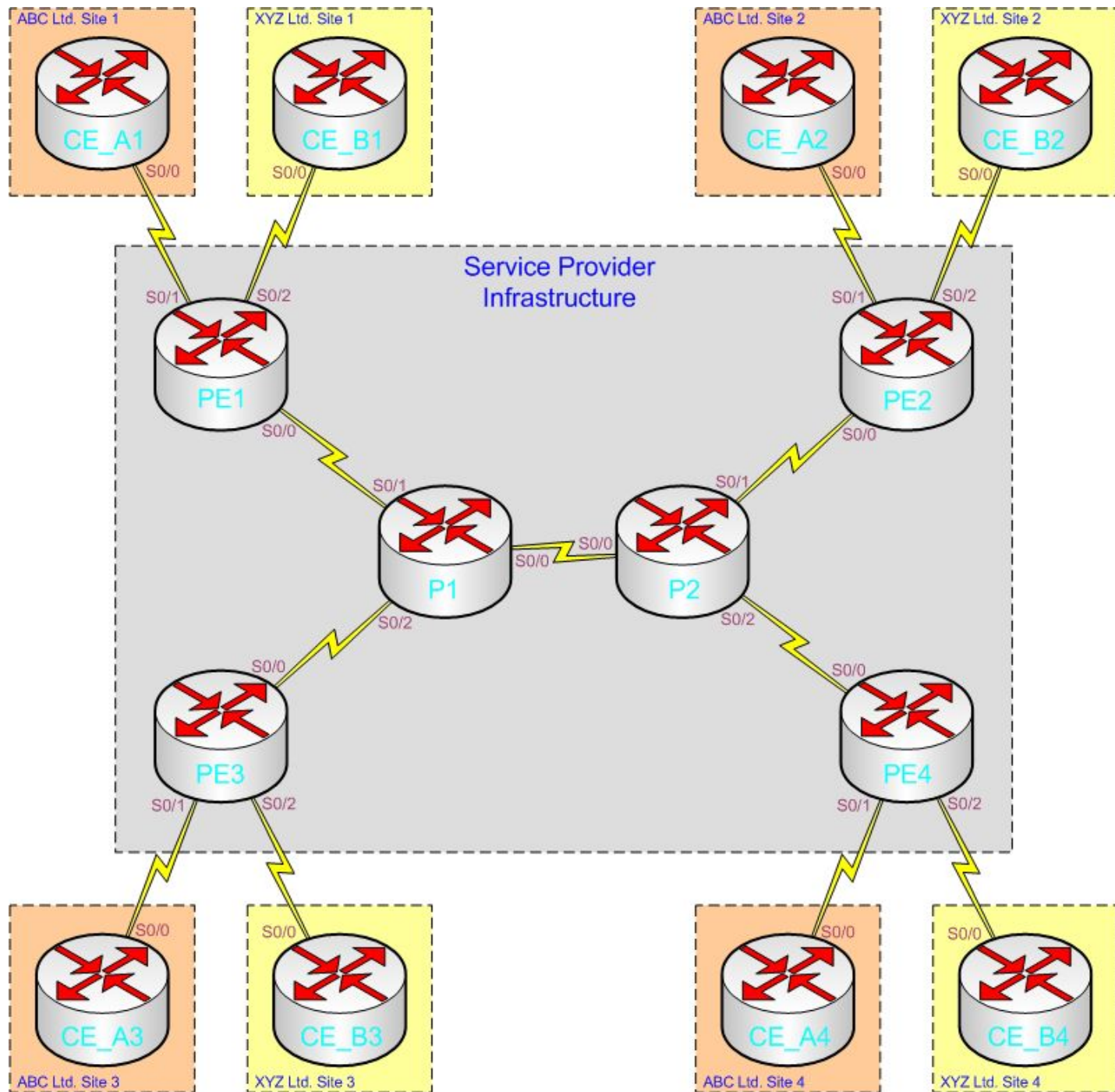


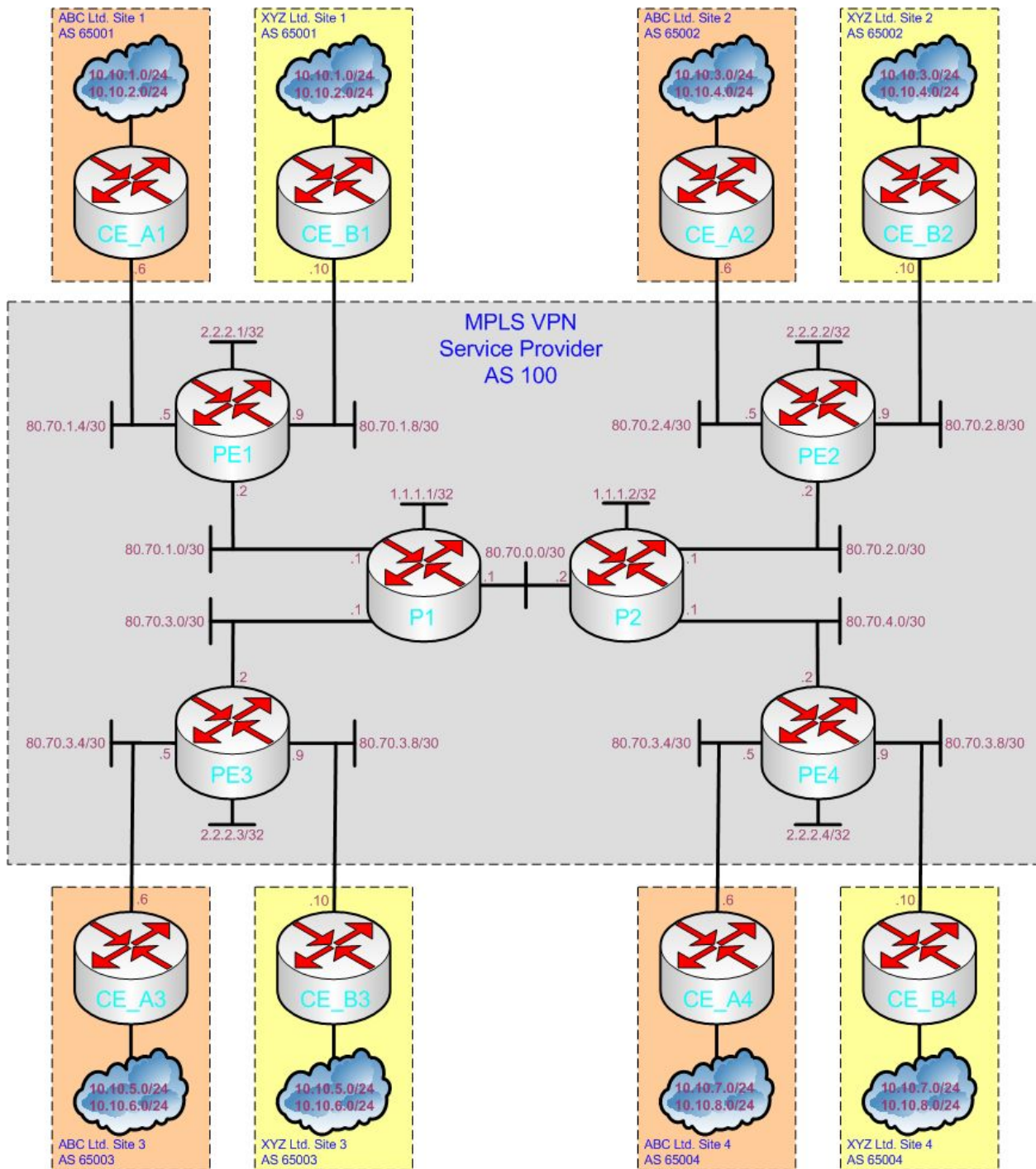
MPLS VPNs

by Richard Bannister

The Topology

- The next two slides display both the physical and logical topology of our simple example network
 - Please study the diagrams carefully before moving on





Laying The Foundations

- Terms that are often heard in close proximity to “MPLS VPN” are “VRFs” and “Multiprotocol BGP”. However, before we begin to look at protocols/features such as these we need to get basic MPLS Unicast IP Forwarding functioning correctly:
 - Step 1 - Enable an IGP on each router within the cloud and verify routing tables are populated correctly
 - BGP (& OSPF) RID reachability is key here
 - Step 2 - Enable MPLS on all ‘provider router to provider router’ interfaces
 - An available label will then be chosen by each router and advertised on all interfaces configured for MPLS using TDP (old default) or LDP (new default & what we’ll use) for each prefix learned via the IGP
 - OSPF will be configured as the IGP in this example
 - Remember. LDP ‘floods’ in all directions so routers will consult the routing table to make a decision on which label path to use for a given prefix (important when loops exist)

Inside The Cloud

Example PE router configuration:

** The first step is to enable MPLS (on all provider routers):

```
ip cef
```

```
mpls ip
```

** In this example we will use the standardised LDP protocol instead of the Cisco proprietary TDP protocol for label advertisement (a change to LDP as the default has recently been made so this command may not be necessary):

```
mpls label protocol ldp
```

```
!
```

** RID loopback:

```
interface Loopback0
```

```
description OSPF & BGP RID
```

```
ip address 2.2.2.1 255.255.255.255
```

```
!
```

** Enable MPLS on interfaces connected to all other PE routers and P routers (not customer side):

```
interface Serial0/0
```

```
description TO P1
```

```
ip address 80.70.1.2 255.255.255.252
```

```
mpls ip
```

```
!
```

** As normal an IGP is configured within the AS -- each learned/connected route on a router will then have a local label assigned and advertised using LDP on all interfaces configured for MPLS. The routing protocol only needs to be enabled on/set to advertise 'provider to provider' interfaces + relevant loopbacks:

```
router ospf 1
```

```
router-id 2.2.2.1
```

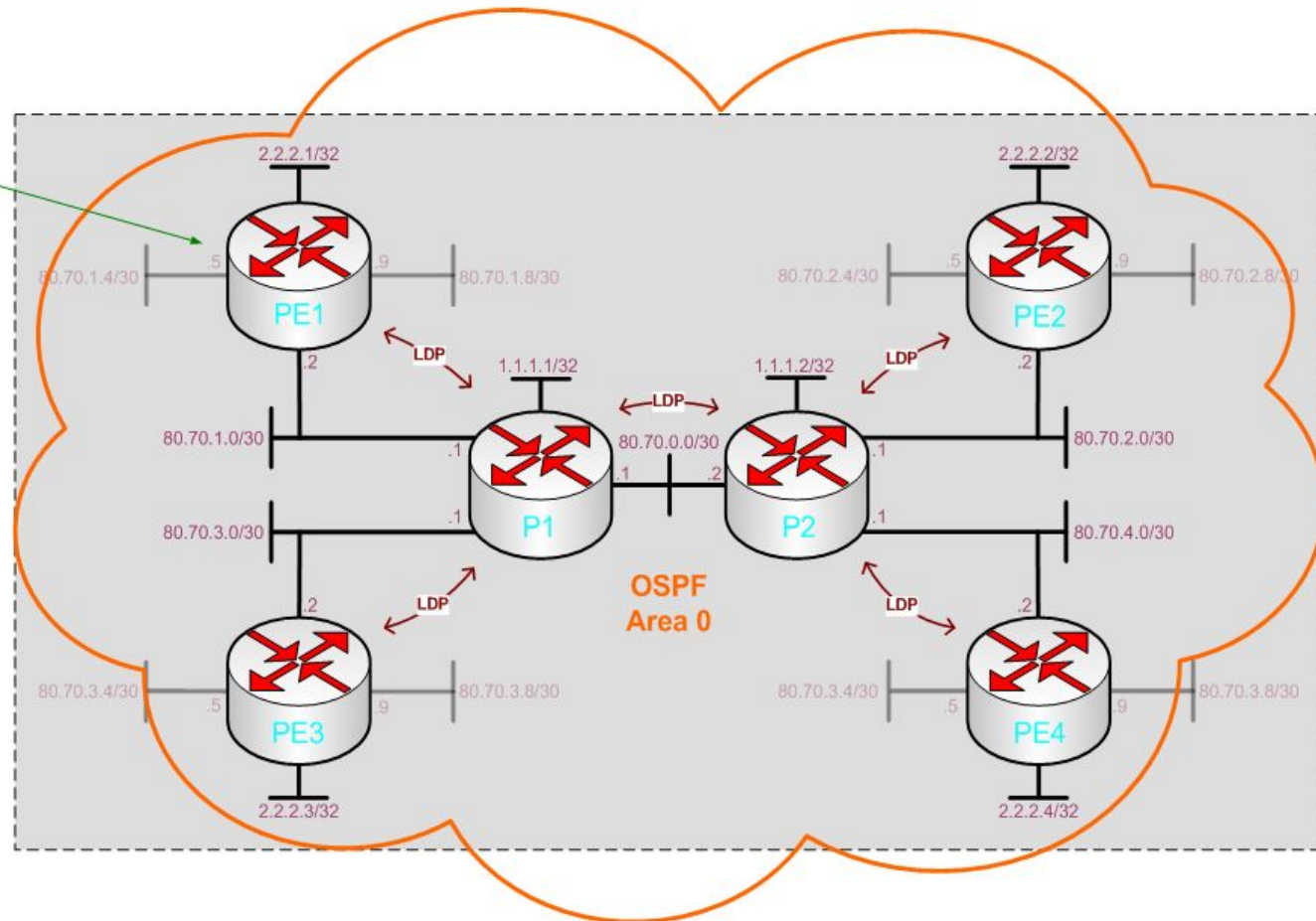
```
log-adjacency-changes
```

```
network 2.2.2.1 0.0.0.0 area 0
```

```
network 80.70.1.2 0.0.0.0 area 0
```

```
!
```

** The configuration above should be amended and applied to all routers within the cloud, this causes MPLS Unicast IP Forwarding to take place and lays the foundations for MPLS VPN



Overlaying MPLS VPNs

- Now that the 'cloud' has basic MPLS IP Unicast Forwarding enabled we can create Virtual Routing and Forwarding (VRF) tables and make use of Multiprotocol BGP (MP-BGP/MBGP) to advertise prefixes held within the VRFs
 - In addition to the standard MPLS label used for Unicast IP Forwarding (now the "outer label" and used to reach a BGP next-hop), an "inner label" will now be added to packets to inform other PE routers of the correct egress VRF tables to place packets in upon receiving them (ready for standard forwarding)
 - This is how a 'VPN' is created – by separating routing information per-customer at the edge of the cloud and ignoring IP headers for forwarding within the cloud

Customer to Provider

- As MPLS is a layer 3 technology a method of communicating routing information between the customer (with possible overlapping prefixes with other customers) and the provider (VRF tables used to separate customers) is required
 - Methods of CE to PE route advertisement include:
 - Static routing
 - RIPv2
 - EIGRP
 - OSPF
 - BGP

Advertisement Flow



How the 10.10.1.0/24 + 10.10.2.0/24 prefixes are advertised from "CE_A1" to "CE_A2"

①

Possible redistribution required
(e.g. the customer IGP = EIGRP & the provider's 'CE to PE routing protocol' choice = RIPv2)

②

Routing information shared between the customer-side 'CE to PE routing protocol' instance and the provider-side 'CE to PE routing protocol' VRF table
(e.g. RIPv2 advertises 10.10.1.0/24 as being reachable via 80.70.1.6)

③

Redistribution = 'CE to PE routing protocol' VRF table into the customer's MBGP VRF table
(next-hop of the BGP RID is set when the route is advertised to MBGP peers)

④

An IBGP session advertises an extended NLRI for 10.10.1.0/24 that distinguishes it from all other advertisements and informs the BGP peer of the VRF it belongs to



⑦

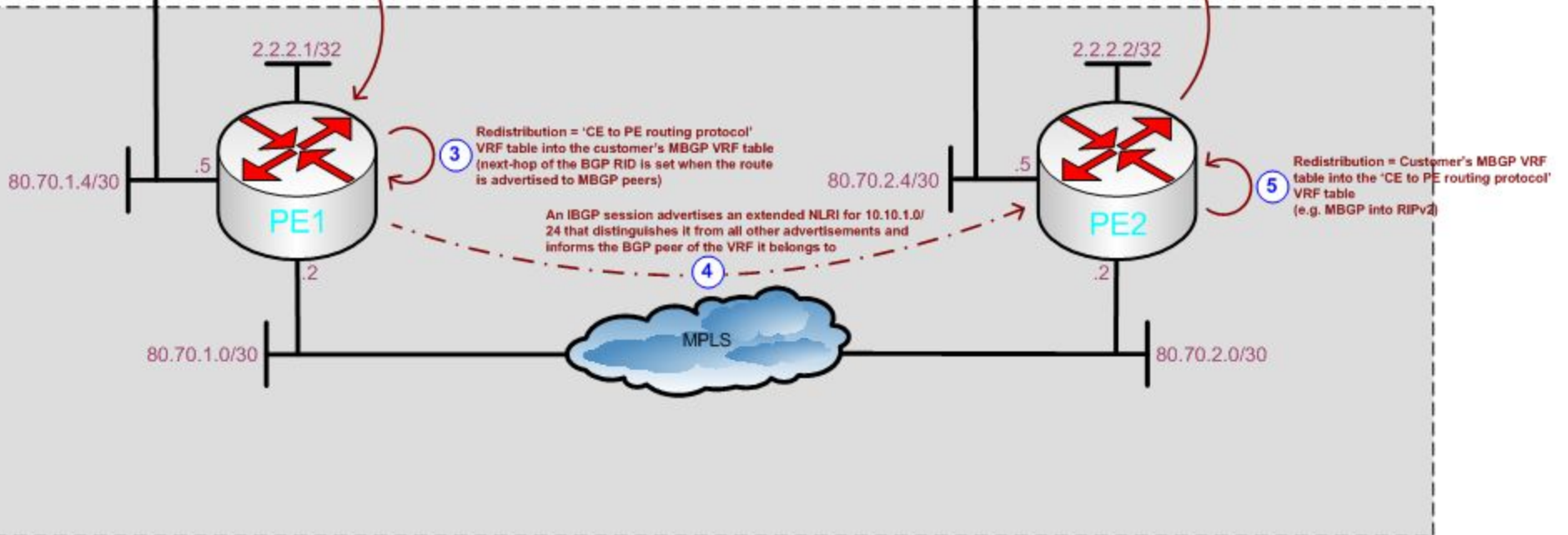
Possible redistribution required
(e.g. RIPv2 into EIGRP)

⑥

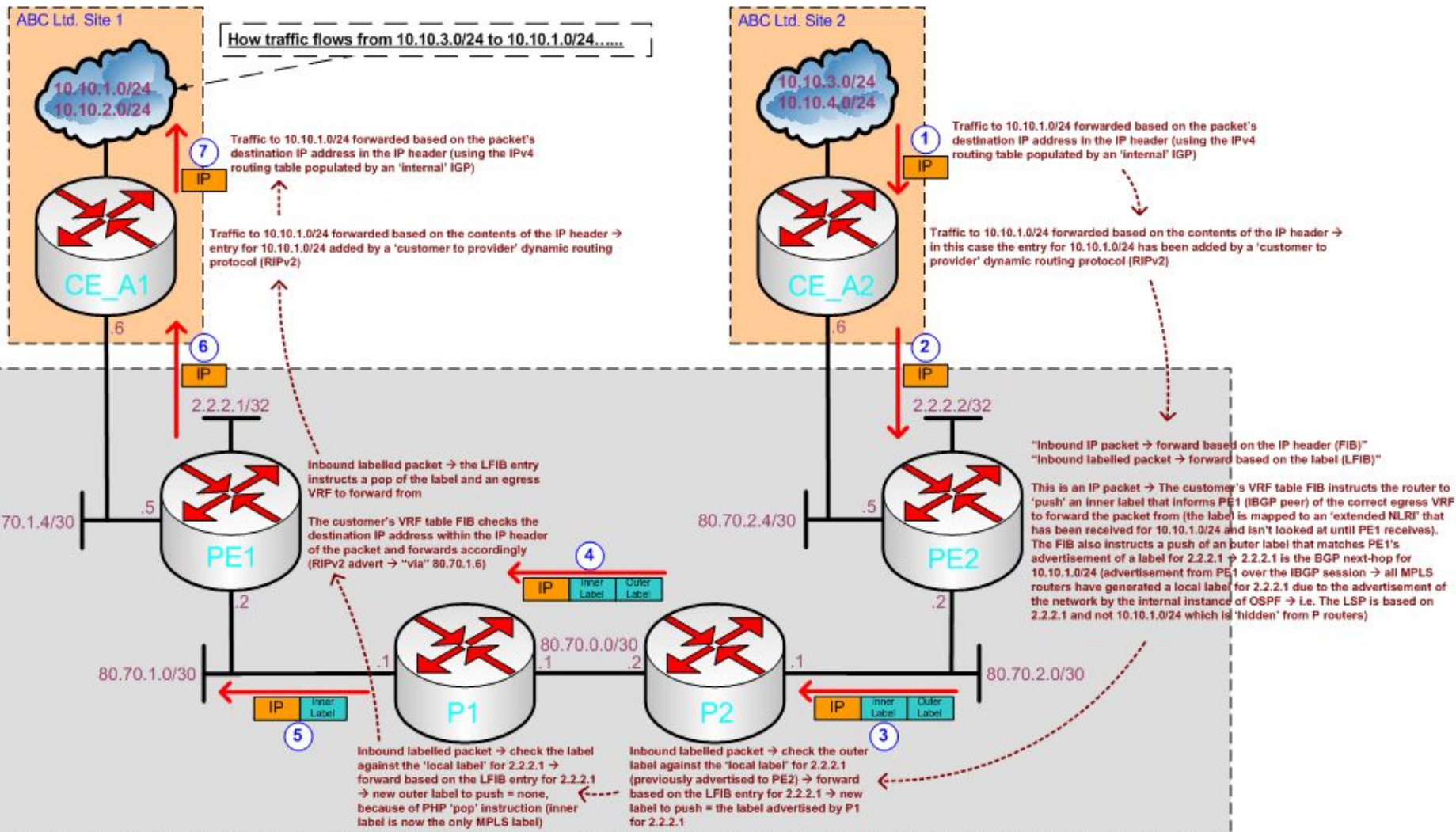
Routing information shared between provider side 'CE to PE routing protocol' VRF table and customer side 'CE to PE routing protocol' instance
(e.g. RIPv2 advertises 10.10.1.0/24 as being reachable via 80.70.2.5)

⑤

Redistribution = Customer's MBGP VRF table into the 'CE to PE routing protocol' VRF table
(e.g. MBGP into RIPv2)



Traffic Flow



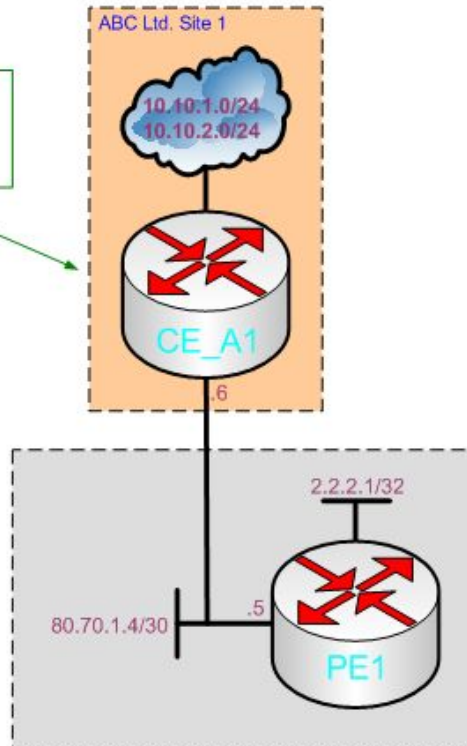
MPLS VPN - CE to PE Static Routing Example

In this portion of the network we will use static routes on the PE1 router (& CE_A1). The routes will then be placed into a VRF table on the PE and advertised using MBGP to other PE routers 'at the other side' of the cloud.

(Additional customer VRF tables can be added by 'duplicating' the configuration with new VRF table names.)

CE_A1:
`ip route 10.10.3.0 255.255.254.0 80.70.1.4`

** No additional configuration required (MPLS VPN is 'transparent' to the customer)



PE1:

** The first task to complete on the PE router is to peer with all other PE routers using Multiprotocol BGP, this involves two stages; standard BGP configuration + MBGP configuration per-peer (remember OSPF is running as an IGP within the cloud = RID reachability + label generation):

```
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 update-source Loopback0
neighbor 2.2.2.3 remote-as 100
neighbor 2.2.2.3 update-source Loopback0
neighbor 2.2.2.4 remote-as 100
neighbor 2.2.2.4 update-source Loopback0
no auto-summary
!
address-family vpnv4
neighbor 2.2.2.2 activate
neighbor 2.2.2.2 send-community extended
neighbor 2.2.2.3 activate
neighbor 2.2.2.3 send-community extended
neighbor 2.2.2.4 activate
neighbor 2.2.2.4 send-community extended
exit-address-family
!
```

** Once the BGP peer relationship has been formed we can then send 'extended NLRIs' over the BGP link. The RD allows PEs to advertise routes as unique prefixes (even if the prefixes overlap) that identify in which VPN the prefix belongs. The RT tells the PE routers which routes should be added to each VRF (provides greater control and possibility for overlapping VPNs):

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

** Add the customer interface to the VPN (remember 'ip address' after VRF!):

```
interface Serial0/1
ip vrf forwarding ABCLtd
ip address 80.70.1.5 255.255.255.252
!
```

** We can now go back to the BGP configuration and define the properties of the customer's MBGP VRF table:

```
router bgp 100
address-family ipv4 vrf ABCLtd
redistribute connected
redistribute static
no synchronization
exit-address-family
!
```

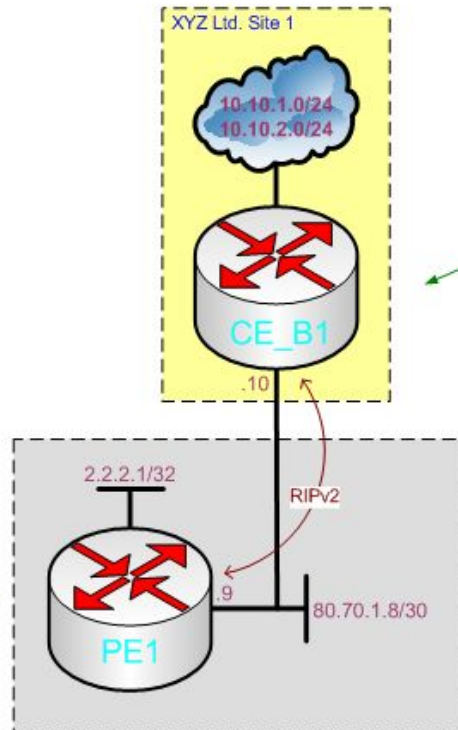
** In this case we are redistributing static routes → so lastly let's add a couple to the 'ABCLtd' VRF table:

```
ip route vrf ABCLtd 10.10.1.0 255.255.255.0 80.70.1.6
ip route vrf ABCLtd 10.10.2.0 255.255.255.0 80.70.1.6
!
```


MPLS VPN - CE to PE RIPv2 Example

Static routing is always an option but the same limitations apply when using static routing with MPLS VPN than they do when using 'normal' static routing. We can use a dynamic routing protocol to advertise a customers routes between a CE and PE but one or two routing protocols have little nuances that need to be noted. All of the routing protocols that are available need to be 'VRF aware'. RIP gives us a good example of a routing protocol + VRF configuration whilst not being overly complicated:

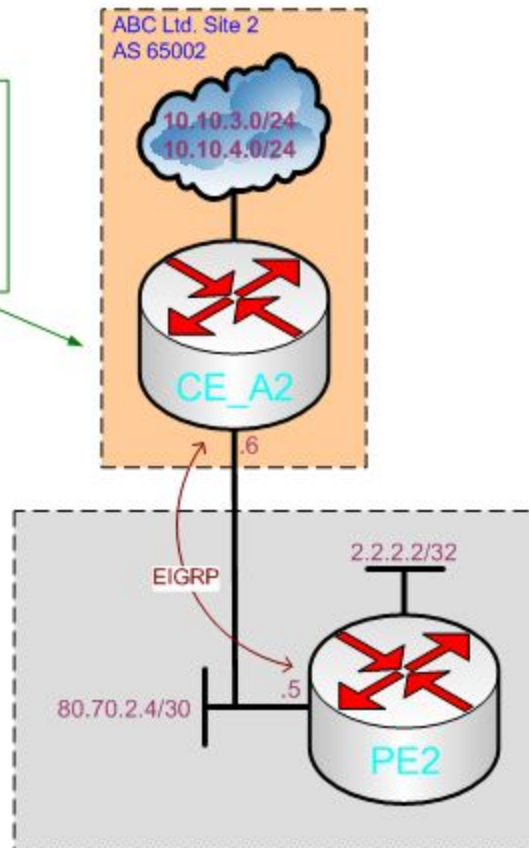
```
PE1:
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 update-source Loopback0
neighbor 2.2.2.3 remote-as 100
neighbor 2.2.2.3 update-source Loopback0
neighbor 2.2.2.4 remote-as 100
neighbor 2.2.2.4 update-source Loopback0
no auto-summary
!
address-family vpnv4
neighbor 2.2.2.2 activate
neighbor 2.2.2.2 send-community extended
neighbor 2.2.2.3 activate
neighbor 2.2.2.3 send-community extended
neighbor 2.2.2.4 activate
neighbor 2.2.2.4 send-community extended
exit-address-family
!
ip vrf XYZLtd
rd 100:2
route-target export 100:2
route-target import 100:2
!
interface Serial0/2
ip vrf forwarding XYZLtd
ip address 80.70.1.9 255.255.255.252
!
** RIPv2 configuration (inc. VRF creation - note, RIPv1 doesn't
support VRFs) for CE to PE route information transfer -> notice the
RIP "network" statement within the VRF only includes the CE_B1 to
PE1 interface's network. Mutual redistribution is performed with
BGP:
router rip
version 2
no auto-summary
!
address-family ipv4 vrf XYZLtd
redistribute bgp 100 metric 2
network 80.0.0.0
no auto-summary
version 2
exit-address-family
!
** We can now go back to the BGP configuration and define the
properties of the customer's MBGP VRF table:
router bgp 100
address-family ipv4 vrf XYZLtd
redistribute connected
redistribute rip
no synchronization
exit-address-family
!
```



```
CE_B1:
** Advertise 'inside' networks and enable RIP on the 'outside'
interface using the 'normal' RIP configuration procedure:
router rip
version 2
network 10.0.0.0
network 80.0.0.0
no auto-summary
!
```

MPLS VPN - CE to PE EIGRP Example

CE_A2:
 ** Advertise 'inside' networks and enable EIGRP on the 'outside' interface using the 'normal' EIGRP configuration procedure:
 router eigrp 50
 network 10.10.3.0 0.0.0.255
 network 10.10.4.0 0.0.0.255
 network 80.70.2.4 0.0.0.3
 no auto-summary
 !



PE2:
 router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 2.2.2.3 remote-as 100
 neighbor 2.2.2.3 update-source Loopback0
 neighbor 2.2.2.4 remote-as 100
 neighbor 2.2.2.4 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor 2.2.2.2 activate
 neighbor 2.2.2.2 send-community extended
 neighbor 2.2.2.3 activate
 neighbor 2.2.2.3 send-community extended
 neighbor 2.2.2.4 activate
 neighbor 2.2.2.4 send-community extended
 exit-address-family
 !
 ip vrf ABCLtd
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 !
 interface Serial0/1
 ip vrf forwarding ABCLtd
 ip address 80.70.2.5 255.255.255.252
 !
 ** EIGRP configuration (inc. VRF creation) for CE to PE route information transfer:
 router eigrp 1
 no auto-summary
 !
 address-family ipv4 vrf ABCLtd
 redistribute bgp 100 metric 10000 100 255 1 1500
 network 10.0.0.0 0.255.255.255
 network 80.70.2.4 0.0.0.3
 no auto-summary
 autonomous-system 50
 exit-address-family
 !
 ** We can now go back to the BGP configuration and define the properties of the customer's MBGP VRF table:
 router bgp 100
 address-family ipv4 vrf ABCLtd
 redistribute connected
 redistribute eigrp 50
 no synchronization
 exit-address-family
 !

MPLS VPN - CE to PE OSPF Example

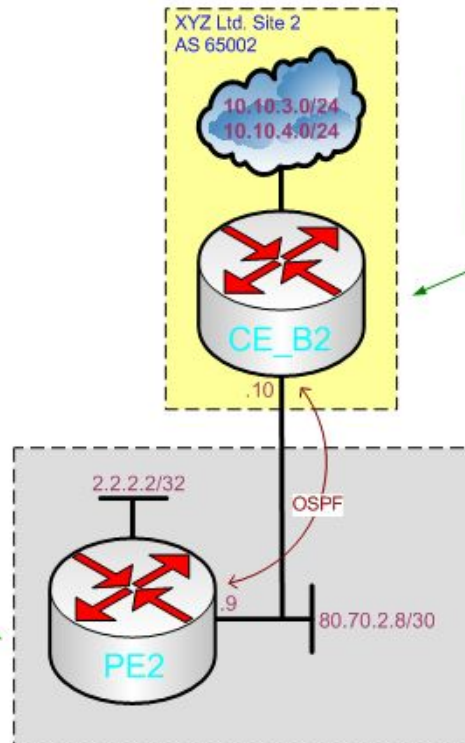
```

PE2:
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 update-source Loopback0
neighbor 2.2.2.3 remote-as 100
neighbor 2.2.2.3 update-source Loopback0
neighbor 2.2.2.4 remote-as 100
neighbor 2.2.2.4 update-source Loopback0
no auto-summary
!
address-family vpnv4
neighbor 2.2.2.2 activate
neighbor 2.2.2.2 send-community extended
neighbor 2.2.2.3 activate
neighbor 2.2.2.3 send-community extended
neighbor 2.2.2.4 activate
neighbor 2.2.2.4 send-community extended
exit-address-family
!
ip vrf XYZLtd
rd 100:2
route-target export 100:2
route-target import 100:2
!
interface Serial0/2
ip vrf forwarding XYZLtd
ip address 80.70.2.9 255.255.255.252
!
** OSPF configuration for CE to PE route information transfer. Note.
OSPF is already running on this router as the provider's IGP so
another instance of OSPF is created by defining a "vrf":
router ospf 2 vrf XYZLtd
log-adjacency-changes
redistribute bgp 100 subnets
network 80.70.2.9 0.0.0.0 area 0
!
** Redistribute OSPF learned routes into the customer's MBGP VRF
table:
router bgp 100
address-family ipv4 vrf XYZLtd
redistribute connected
redistribute ospf 2 vrf XYZLtd
no synchronization
exit-address-family
!

** When CE_B2 receives adverts originated by other CE routers in
its VPN they will be imported using Type3 LSAs even though the
router believes the prefixes to be in the same area (0) - to get
around this 'problem' a provider can create a special tunnel called a
'sham link' (between the edge-LSRs - so, this is required at both
sides - reverse the configuration...):
interface Loopback 1
ip vrf forwarding XYZLtd
ip address 192.168.255.1 255.255.255.255
!
router ospf 2 vrf XYZLtd
area 0 sham-link 192.168.255.1 192.168.255.2 cost 10
!
** The end points of the tunnel must be reachable within the VPN -
see "redistribute connected" under the BGP configuration

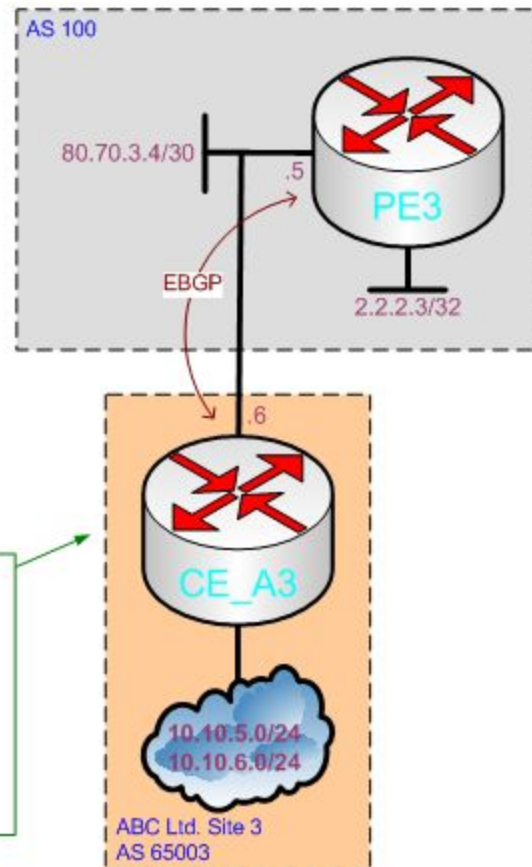
```

What if we are already running OSPF on the PE?.....



CE_B2:
 ** Enable OSPF using the 'normal' OSPF configuration procedure:
 router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
 network 80.70.2.10 0.0.0.0 area 0
 !

MPLS VPN - CE to PE BGP Example



CE_A3:
**** Enable BGP and 'inject' prefixes to advertise using the 'normal' BGP configuration procedure:**
 router bgp 65003
 no synchronization
 bgp log-neighbor-changes
 network 10.10.5.0 mask 255.255.255.0
 network 10.10.6.0 mask 255.255.255.0
 neighbor 80.70.3.5 remote-as 100
 neighbor 80.70.3.5 update-source Serial1/0
 no auto-summary
 !

PE3:
 router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 2.2.2.3 remote-as 100
 neighbor 2.2.2.3 update-source Loopback0
 neighbor 2.2.2.4 remote-as 100
 neighbor 2.2.2.4 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor 2.2.2.2 activate
 neighbor 2.2.2.2 send-community extended
 neighbor 2.2.2.3 activate
 neighbor 2.2.2.3 send-community extended
 neighbor 2.2.2.4 activate
 neighbor 2.2.2.4 send-community extended
 exit-address-family
 !
 ip vrf ABCLtd
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 !
 interface Serial0/1
 ip vrf forwarding ABCLtd
 ip address 80.70.3.5 255.255.255.252
 !
**** To run BGP with the customer simply define a neighbour within the customer's BGP VRF table:**
 router bgp 100
 address-family ipv4 vrf ABCLtd
 redistribute connected
 neighbor 80.70.3.6 remote-as 65003
 neighbor 80.70.3.6 activate
 no synchronization
 exit-address-family
 !