MPLS Notes

P - Provider (core/backbone)
PE- Provider Edge - customer facing
CE - Customer Edge
 - those 3 from VPN terminology

LSP - Label Switched Path
LER - Label Edge Router (ingress node)
LSR Label Switching Router (transit nodes)
Egress Node - end of the LSP (destination)

The LSP refers to a MPLS-wrapped unidirectional path between a pair of routers over the larger network. Return traffic will have it's own LSP

Termination may also be L2VPN (pseudowire), L3VPN VPLS

MPLS L2VPN (pseudowire)
2 historical methods of signalling:
 - LDP-signalled - "Draft Martini" - Simpler, common
 - BGP-signalled "Draft Kompella" - More complex, has autodiscovery and multipoint support
        VLL - Virtual Leased Lines - emulated point-to-point circuit delivered over MPLS (IETF PWE3 Working Group)
         - interconnect 2 different types of media (FR to Eth)
         - migrate transport from legacy (ATM to MPLS)
         - unknown payload hard to load balance - can't see IP header inside)

MPLS L3VPN
 - IP-based, VRFs on (BGP) edge routers
 - Load balancing-friendly (exposed IP headers)
 - significant load to service provider infrastructure
 - often in enterprise

MPLS labels are link-local
Needs signaling protocols:
 - LDP - Label Distribution Protocol - no traffic engineering!
 - RSVP-TE - Resource RSVP protocol with Traffic Engineering
        - Complex, more overhead
        - Most MPLS needs both, LDP for transport inside RSVP-TE

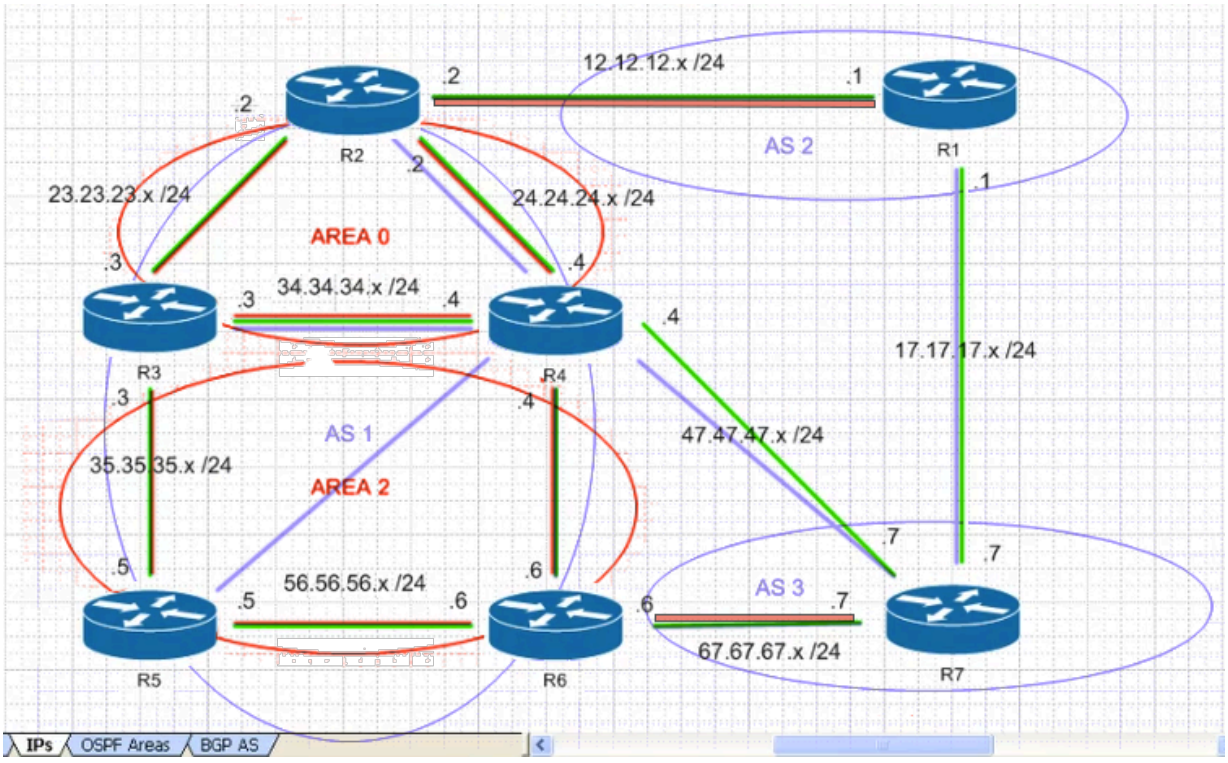
Stacked MPLS labels/ LSPs
MPLS labels are stacked/ encapsulated with another MPLS label. At first destination one is torn off and the traffic continues to the next layer's egress

Frames are tagged with a LSP on one network segment en route to another MPLS network at destination. This traffic is labeled with a "bypass" LDP layer

unlabled frame -- LER-----PE -------%=====================%--------PE
PE -------LER==P====P=====P====LER--------PE------LER
----unlabled frame

Unlabled hits a Label Edge Router. Gets tagged with its first label/ label switched path
 - this first LSP traffic will use LDP label dist protocol.
Traffic tagged with "first layer" MPLS LSP hits a provider edge LER that adds a second layer of MPLS tagging
 - this 2nd tag is for transit through the provider using RSVP-TE
 - at the RSVP-TE, the LER strips off it's layer of MPLS tagging, leaving the one MPLS label/LSP on the frames
Eventually the innermost LSP tagging is stripped off by the destination LER to hand over unlabeled frames

The RSVP-TE "tunnel" cuts down on overhead since the inner LSP traffic isn't relevant to transport (ins't looked at necessarily).
RSVP-TE traffic can prioritize routes to
 - get best bandwidth/ a shorter path
 - avoid outages or areas with insufficient bandwidth (redirection)
 - employ selective routing
 - a de facto QoS that doesn't drop traffic (assigned priority)



Red is OSPF, blue is BGP AS, green is simply our IP links
Every interface with an OSPF neighbor is also going to have MPLS enabled (TDP or

LDP neighbor adjacencies)
[LDP is actually based on Cisco's proprietary TDP (Tag Distribution Protocol)]
TE is implemented in OSPF area 0
It's exchanging MPLS across AS boundaries, (e.g. R7 exports it's prefixes for R6 to put into a VRF)
R6 can then advertise prefixes across it's AS to R2
R2 can then advertise those prefixes from that VRF to R1 via VPNv4 BGP announcements

[virtual routing and forwarding (VRF) is a technology that allows multiple instances of a routing table to co-exist within the same router at the same time. Because the routing instances are independent, the same or overlapping IP addresses can be used without conflicting with each other. Network functionality is improved because network paths can be segmented without requiring multiple routers.]

[VPNv4 is the protocol used between PE routers to exchange customers (CE) prefixes (prepended RD, and send RT BGP community) and labels information.

[A prefix is an IP address and network mask. It can represent an entire network, a subset of a network, or a single host route]

---
bgp vpnv4 is a requirement when deployment L3 mpls vpn.

Re: two commands under bgp, the address-family vpnv4 and address-family ipv4. What is the advantage or disadvantage if you're not using address-family vpnv4 in MPLS VPNs? Will it also work for VRF-Lite?
A: Multiprotocol BGP supports carrying multiprotocol prefixes over the BGP. Its like saying it not only carries the IPv4 prefixes but can carry IPv6, IPx prefixes too. But by default(I guess), BGP carry IPv4 prefixes.
BGP uses capability negotiation in order to confirm that the peer supports the required capabilities. These capabilities include different values for AFI and SAFI. Peering router (both) must support a capability to use it. It is advertised in Open messages,
And the main benefit of address-family is that one can associate a different routing policy for same peer for different protocol prefixes.

why is it that vpnv4 address family is not available in cisco 3550 or 3560s?
because they only support VRF Lite not full blown MPLS VPNs.

--- See output of these in 2.2. PDF for MPLS forwarding which needs CEF enabled
show ip cef summary
show ip cef
debug ip cef table

ip cef
no ip cef

In example, CEF is brought up on R2 first. It builds it's CEF table by grabbing the prefixes from OSPF

R2# show ip cef

Prefix	Next Hop	Interface
0.0.0.0/0	drop	Null0 (default route handler entry)
0.0.0.0/32	receive	Ethernet0/1
2.2.2.2/32	receive	Ethernet0/2
3.3.3.3/32	23.23.23.3	Ethernet0/1
4.4.4.4/32	24.24.24.4	Ethernet0/2
5.5.5.5/32	23.23.23.3	Ethernet0/3
6.6.6.6/32	24.24.24.4	Loopback22
12.12.12.0/24	attached	Ethernet0/1
12.12.12.0/32	receive	Ethernet0/1
	<SNIP>	
55.55.55.0/24	23.23.23.3	Ethernet0/1
56.56.56.0/24	24.24.24.4	Ethernet0/2
224.0.0.0/4	drop	
224.0.0.0/24	recieve	
255.255.255.255/32	receive	

Turn on LDP on R2, then override to be TDP with R3

R2
show mpls ldp neighbor
show mpls ldp bindings
! is it already running on the router? See baseline

debug mpls ldp peer state-machine
 -- info about state transitions for LDP sessions
LDP manages peer sessions by means of the following two coupled state machines:
- A low-level state machine that deals with session establishment and shutdown.
- A high-level state machine that deals with setting up and shutting down label advertisement

debug mpls ldp messages sent
 -- display contents of LDP messages sent and received from LDP peers.
LDP requires periodic transmission of keepalive messages. If you do not specify the "all" option, periodic keepalive messages are not displayed

Turn it on (globally):
mpls label protocol ldp

There will be no debug msgs or output to these commands until also turned on at the interface level:
show mpls ldp neighbor
show mpls ldp bindings

R2
interface e0/1
  mpls label protocol ?
  mpls label protocol tdp
---- you could say "both" (instead of TDP or LDP) if you need a multiaccess interface
R3
conf t
interface e0/2
  mpls label protocol tdp

R2
show mpls ldp neighbor
show mpls ldp bindings
show mpls interfaces

Still no debug info! You have to do configure the MPLS router ID and turn on MPLS. Hardcode it to make sure its an IP of your choosing
(making sure it is an IP address that will not be prevented from being advertised to the neighboring router)

mpls ldp router-id loopback 0 force
- force makes it take effect immediately (instead of on reboot or something)

You still won't see any significant info until you finally turn on MPLS on the interfaces, since we just were laying the framework so far:

 interface e0/1
  mpls ip
interface e0/2
  mpls ip

After doing this debug is spitting out info about MPLS traffic between this router (R2) and R3 and R4

show mpls ldp neighbor
Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 3.3.3.3.38735 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 16/15; Downstream
        Up time: 00:00:23
        LDP discovery sources:

```
            Ethernet0/1, Src IP addr: 23.23.23.3
        Addresses bound to peer LDP Ident:
            35.35.35.3      3.3.3.3         33.33.33.33         34.34.34.3
            23.23.23.3

Peer LDP Ident: 4.4.4.4:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 4.4.4.4.18235 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 16/16; Downstream
        Up time: 00:00:22
        LDP discovery sources:
            Ethernet0/2, Src IP addr: 24.24.24.4
        Addresses bound to peer LDP Ident:
            46.46.46.4      4.4.4.4         44.44.44.44         34.34.34.4
            24.24.24.4      47.47.47.4
```

show mpls ldp bindings - shows bindings between prefixes and tags, e.g.:

```
tib entry: 2.2.2.2/32, rev 6
      local binding:  tag: imp-null
      remote binding: tsr: 3.3.3.3:0, tag: 21
      remote binding: tsr: 4.4.4.4:0, tag: 17
tib entry: 3.3.3.3/32, rev 8
      local binding:  tag: 18
      remote binding: tsr: 3.3.3.3:0, tag: imp-null
      remote binding: tsr: 4.4.4.4:0, tag: 18
tib entry: 4.4.4.4/32, rev 10
      local binding:  tag: 19
      remote binding: tsr: 3.3.3.3:0, tag: 22
remote binding: tsr: 4.4.4.4:0, tag: imp-null tib entry: 5.5.5.5/32, rev 12
      local binding:  tag: 20
      remote binding: tsr: 3.3.3.3:0, tag: 17
      remote binding: tsr: 4.4.4.4:0, tag: 21
```

show mpls interfaces

Interface	IP	Tunnel	Operational
Ethernet0/1	Yes (ldp)	No	Yes
Ethernet0/2	Yes (ldp)	No	Yes

Limiting Label Distribution

Failed attempt - filter policy on R3 and R4 of what is advertised to R2 (says is documented but never made it work)

```
show mpls ip bind 5.5.5.5 32
show mpls ip bind 6.6.6.6 32
```

```
R2#show mpls ip bind 5.5.5.5 32
  5.5.5.5/32
        in label:     20
        out label:    21         lsr: 3.3.3.3:0         inuse
        out label:    20         lsr: 4.4.4.4:0
R2#show mpls ip bind 6.6.6.6 32
  6.6.6.6/32
        in label:     21
        out label:    22         lsr: 3.3.3.3:0
        out label:    21         lsr: 4.4.4.4:0         inuse
```

--- On R3, R4 put this in
access-list 1 permit 56.56.56.0 0.0.0.255
access-list 2 permit host 2.2.2.2
!
mpls ldp advertise-labels for 1 to 2
--- let labels for 56.56.56.0 be advertised to 2.2.2.2)

--- On R2, clear the peers to rebuild it:
clear mpls ldp neighbor *
--- Check it:
show mpls ip bind 5.5.5.5 32
show mpls ip bind 6.6.6.6 32
--- And it is the same!!

But THIS works -
Filter what R2 will accept from R3 and R4

On R2
ip access-list standard ACCEPT
  permit 56.56.56.0 0.0.0.255

---------- From these neighbors, only accept labels referred to in this access list
mpls ldp neighbor 3.3.3.3 labels accept ACCEPT
mpls ldp neighbor 4.4.4.4 labels accept ACCEPT

```
R2#show mpls ip bind 5.5.5.5 32
  5.5.5.5/32
        in label:     20
R2#show mpls ip bind 6.6.6.6 32
  6.6.6.6/32
        in label:     21
```

show mpls ldp neighbor
show mpls ldp binding
debug mpls ldp messages received
show mpls ldp neighbor

show mpls ip bind 5.5.5.5 32
show mpls ip bind 6.6.6.6 32
!

Traffic Engineering

MPLS Traffic Engineering Network
Essential for service provider backbones - high use of transmission capacity, and resilient- can withstand link or node failures.
TE capabilities integrated into Layer 3 optimizes IP traffic routing, given the constraints imposed by backbone capacity and topology.
Routes traffic flows across a network based on the resources the traffic flow requires and the available resources
Employs "constraint-based routing," traffic flow is the shortest path that meets the resource constraints of the traffic flow.
The flow has bandwidth requirements, media requirements, a priority versus other flows, and so on.
Gracefully recovers from link or node failures that change the topology of the backbone by adapting to the new set of constraints.

MPLS RSVP TE tunnels establish unidirectional label switching paths.
Serves similar purpose as label distribution using LDP - establishing label switched path that ensures frame delivery from ingress to egress router, but with additional features:
Possibility to establish label switching path using either full or partial explicit route;
Constraint based LSP establishment - label switching path is established over links that fulfill requirements, such as bandwidth and link properties.
MPLS RSVP TE is based on RSVP protocol with extensions introduced by RFC 3209 that adds support for explicit route and label exchange.

In this example R3 and R4 are already set up for TE

R2
show ip cef
show mpls traffic-eng topology brief
show mpls traffic-eng link-management summary
!
debug mpls traffic-eng areas
debug mpls traffic-eng link-management events
!
conf t
ip cef
--- enable the routing protocol (OSPF) to be able to handle MPLS-TE
--- hardcode the RID for TE to loopback
router ospf 2
 mpls traffic-eng area 0
 mpls traffic-eng router-id lo0

```
!
 mpls traffic-eng tunnels
--- Enable globally, then enable on ints
int e0/1
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ------5Mbps
int e0/2
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
!
show mpls traffic-eng topology brief
show mpls traffic-eng link-management summary
```

```
NetStepByStep-R2#show mpls traffic-eng topology brief
My_System_id: 2.2.2.2, Globl Link Generation 12
Signalling error holddown: 10 sec

IGP Id: 2.2.2.2, MPLS TE Id:2.2.2.2 Router Node
      link[0 ]:DR Intf Address: 23.23.23.3, gen:12
          frag_id 0, Intf Address:23.23.23.2
          TE metric:1, IGP metric:1, attribute_flags:0x0

      link[1 ]:DR Intf Address: 24.24.24.4, gen:12
          frag_id 1, Intf Address:24.24.24.2
          TE metric:1, IGP metric:1, attribute_flags:0x0

IGP Id: 3.3.3.3, MPLS TE Id:3.3.3.3 Router Node
      link[0 ]:DR Intf Address: 23.23.23.3, gen:8
          frag_id 0, Intf Address:23.23.23.3
          TE metric:1, IGP metric:1, attribute_flags:0x0

      link[1 ]:DR Intf Address: 34.34.34.4, gen:8
          frag_id 1, Intf Address:34.34.34.3
          TE metric:1, IGP metric:1, attribute_flags:0x0

IGP Id: 4.4.4.4, MPLS TE Id:4.4.4.4 Router Node
      link[0 ]:DR Intf Address: 34.34.34.4, gen:9
NetStepByStep-R2#show mpls traffic-eng link-management summary
System Information::
    Links Count:          2
    Flooding System:      enabled
IGP Area ID::  ospf area 0
    Flooding Protocol:    OSPF
    Flooding Status:      data flooded
    Periodic Flooding:    enabled (every 180 seconds)
    Flooded Links:        2
    IGP System ID:        2.2.2.2
    MPLS TE Router ID:    2.2.2.2
    IGP Neighbors:        2
Link ID::  Fa0/1 (23.23.23.2)
    Link Status:
      Physical Bandwidth:   100000 kbits/sec
      Max Res Global BW:    5000 kbits/sec (reserved: 0% in, 0% out)
      Max Res Sub BW:       0 kbits/sec (reserved: 100% in, 100% out)
      MPLS TE Link State:   MPLS TE on, RSVP on, admin-up, flooded
      Inbound Admission:    reject-huge
      Outbound Admission:   allow-if-room
      Admin. Weight:        1 (IGP)
      IGP Neighbor Count:   1
Link ID::  Fa1/0 (24.24.24.2)
    Link Status:
```

TE Tunnels
Tunneling traffic from R2 to R3 through R4

R2
show mpls traffic-eng tunnels
show ip ospf neighbor ---use to verify that OSPF is seeing tunnel companion as a connected neighbor
show ip route 3.3.3.3 ---display int that connects to R3
!
debug mpls traffic-eng tunnels events
!
conf t
ip explicit-path name R2-R4-R3
 next-address 24.24.24.4 --- R4 int that connects to R2
 next-address 34.34.34.3 --- R3 int that connects to R4
 next-address 3.3.3.3 --- R3's loopback int

interface tunnel 23 --- this is a UNIDIRECTIONAL tunnel!
 ip unnumber lo0
 tunnel destination 3.3.3.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce --- tell IGP to use the tunnel in its enhanced SPF calculation "it isn't in OSPF but treated like it is"
 tunnel mpls traffic-eng bandwidth 2000 --- last config RSVPd 5Mbit BW, this will take up 2Mbit of that
 tunnel mpls traffic-eng path-option 1 explicit name R2-R4-R3
!
show mpls traffic-eng tunnels
show ip ospf neighbor
show ip route 3.3.3.3

R3
show mpls traffic-eng tunnels
show ip ospf neighbor
show ip route 2.2.2.2

 ip unnumber lo0
--- enable IP processing on an interface without assigning it an explicit IP address. The ip unnumbered interface can "borrow" the IP address of another interface already configured on the router, which conserves network and address space.

```
NetStepByStep-R2#show mpls traffic-eng tunnels
Name: NetStepByStep-R2_t23                (Tunnel23) Destination: 3.3.3.3
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit R2-R4-R3 (Basis for Setup, path weight 2)
  Config Parameters:
    Bandwidth: 2000     kbps (Global)  Priority: 7  7   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 2000     bw-based
    auto-bw: disabled
  InLabel  :  -
  OutLabel : FastEthernet1/0, 23
  RSVP Signalling Info:
       Src 2.2.2.2, Dst 3.3.3.3, Tun_Id 23, Tun_Instance 2
    RSVP Path Info:
      My Address: 24.24.24.2
      Explicit Route: 24.24.24.4 34.34.34.4 34.34.34.3 3.3.3.3
      Record Route:  NONE
      Tspec: ave rate=2000 kbits, burst=1000 bytes, peak rate=2000 kbits
    RSVP Resv Info:
NetStepByStep-R2#show ip route 3.3.3.3
Routing entry for 3.3.3.3/32
  Known via "ospf 2", distance 110, metric 2, type intra area
  Last update from 3.3.3.3 on Tunnel23, 00:00:24 ago
  Routing Descriptor Blocks:
  * 3.3.3.3, from 3.3.3.3, 00:00:24 ago, via Tunnel23
      Route metric is 2, traffic share count is 1

NetStepByStep-R2#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
4.4.4.4           1   FULL/DR         00:00:29    24.24.24.4      FastEthernet1/0
3.3.3.3           1   FULL/DR         00:00:36    23.23.23.3      FastEthernet0/1
```

MPLS VPN

- reduce customer networking complexity, costs and totally do away with the requirement of in-house technical work force.
- managing point-to-point circuits for each office using leased lines replaced with 1 connection from office to PE router.
- IP VPN for MPLS) allows a service provider to deploy scalable IPv4 L3 VPN backbone services.

BGP is used to set the path for MPLS packet forwarding (as well as AS traversal) uses iBGP between PE and PE in BGP/MPLS VPN.
PE router receives IP prefix from CE and add 8 bytes RD (Route Distinguisher) to the IP prefix .
VPN is established with defining VRF that is consisted of route distinguisher (RD) and route target (RT).

Configure VRFs on R2 and R6

R2
show ip vrf VPN-A
show ip vrf interfaces
show run interface e0/3

--- Our VRF is named "VPN-A" here
--- sh run int to get the IP address, since when configuring a VRF on an int, it will remove the IP from the global routing table to put in (one of) the VRF tables. Doing this before adding to a VRF ensures you have the IP address noted (plus the subnet it belongs to)

ip vrf VPN-A
 rd 2.2.2.2:2
        --- route descriptor/ route-distinguisher - it is coming from R2 so it gets it's loopback
        --- is added to the beginning of the customer's IPv4 prefixes to change them into globally unique VPN-IPv4 prefixes
 route-target export 2.2.2.2:2   --- exporting the VPNv4 target address advertised out. Specifies *VPN route-target export communities*
 route-target import 6.6.6.6:6   --- similarly imported to the VRF from BGP VPNv4 routes. Specifies *VPN route-target import communities*

--- now, associate the interface to the VRF
interface e0/3
 ip vrf forwarding VPN-A
 ip add 12.12.12.2 255.255.255.0

--- As previously mentioned we have to re-add the IP since it will zap it like this:

```
NetStepByStep-R2(config-if)# ip vrf forwarding VPN-A
% Interface FastEthernet0/0 IP address 12.12.12.2 removed due to enabling VRF VPN-A
NetStepByStep-R2(config-if)# ip add 12.12.12.2 255.255.255.0
```

R6 - generally the same as above, reversed appropriately
show ip vrf VPN-A
show ip vrf interfaces
show run interface e0/3

ip vrf VPN-A
 rd 6.6.6.6:6
 route-target export 6.6.6.6:6
 route-target import 2.2.2.2:2

interface e0/3
 ip vrf forwarding VPN-A
 ip address 67.67.67.6 255.255.255.0

--- disassociate a VRF with the "no ip vrf forwarding VPN-A" form

show ip vrf VPN-A
show ip vrf interfaces

```
NetStepByStep-R6#show ip vrf VPN-A
  Name                             Default RD          Interfaces
  VPN-A                            6.6.6.6:6           Fa1/0
NetStepByStep-R6#show ip vrf interfaces
Interface              IP-Address      VRF                              Protocol
Fa1/0                  67.67.67.6      VPN-A                            up
```

Getting rid of IPv4 Unicast (BGP)
These changes were already made on other routers)

R2
show ip bgp sum
show run | be router bgp
---- check out your stuff first. BGP adjacency went down from when we switched interface to VRF before.

router bgp 1
 no bgp default ipv4-unicast
---- this is on by default and we are turning it off. Now there should be specific address-family IPv4 entries for BGP adjacent neighbors

show ip bgp sum
show run | be router bgp

PE-to-PE Routing Sessions
Set up R2 and R6 for this, then look at the big picture since rest of the network has already been configured
R2 still has adjacency issues from VRF application. R4 is still good to go.

On R2
show ip bgp sum
debug ip bgp vpnv4 unicast --- display debugging info related to importing of BGP paths into a VRF instance table
!
router bgp 1
 address-family vpnv4
 neighbor 4.4.4.4 activate

--- The address-family vpnv4 command replaces the match nlri and set nlri commands
--- For all other address families, address exchange is disabled by default.
--- You can explicitly activate the default command using the appropriate address family submode.

--- Neighbor goes down and comes back up when we do this since it is adding a feature to the adjacency (vpnv4)

```
NetStepByStep-R2(config)#router bgp 1
NetStepByStep-R2(config-router)# address-family vpnv4
NetStepByStep-R2(config-router-af)# neighbor 4.4.4.4 activate
NetStepByStep-R2(config-router-af)#
*Mar  1 02:24:43.951: %BGP-5-ADJCHANGE: neighbor 4.4.4.4 Down Address family activated
*Mar  1 02:24:46.187: %BGP-5-ADJCHANGE: neighbor 4.4.4.4 Up
```

---- Activate on R6 too:
show ip bgp sum
show run l be router bgp
!
router bgp 1
 address-family vpnv4
 neighbor 4.4.4.4 activate
!
show ip bgp sum

```
NetStepByStep-R6#show ip bgp sum
BGP router identifier 6.6.6.6, local AS number 1
BGP table version is 24, main routing table version 24
5 network entries using 585 bytes of memory
5 path entries using 260 bytes of memory
3/2 BGP path/bestpath attribute entries using 372 bytes of memory
3 BGP rrinfo entries using 72 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1289 total bytes of memory
BGP activity 9/4 prefixes, 15/10 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
4.4.4.4         4     1     181     172       24    0    0 00:00:16        4
67.67.67.7      4     3     146     155        0    0    0 00:16:38 Idle
```

R2
show ip bgp sum
R4
sh run l be router bgp

```
NetStepByStep-R4#sh run | be router bgp
router bgp 1
 bgp router-id 4.4.4.4
 no bgp default ipv4-unicast
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 2.2.2.2 remote-as 1
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 3.3.3.3 remote-as 1
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 5.5.5.5 remote-as 1
 neighbor 5.5.5.5 update-source Loopback0
 neighbor 6.6.6.6 remote-as 1
```

```
neighbor 6.6.6.6 update-source Loopback0
!
address-family ipv4
 neighbor 2.2.2.2 activate
 neighbor 2.2.2.2 send-community both
 neighbor 2.2.2.2 route-reflector-client
 neighbor 3.3.3.3 activate
 neighbor 3.3.3.3 send-community both
 neighbor 3.3.3.3 route-reflector-client
 neighbor 5.5.5.5 activate
 neighbor 5.5.5.5 send-community both
 neighbor 5.5.5.5 route-reflector-client
 neighbor 6.6.6.6 activate
 neighbor 6.6.6.6 send-community both
 neighbor 6.6.6.6 route-reflector-client
 no auto-summary
 no synchronization
 network 44.44.44.0 mask 255.255.255.0
exit-address-family
!
address-family vpnv4
 neighbor 2.2.2.2 activate
 neighbor 2.2.2.2 send-community extended
 neighbor 2.2.2.2 route-reflector-client
 neighbor 3.3.3.3 activate
 neighbor 3.3.3.3 send-community extended
 neighbor 3.3.3.3 route-reflector-client
 neighbor 5.5.5.5 activate
 neighbor 5.5.5.5 send-community extended
 neighbor 5.5.5.5 route-reflector-client
 neighbor 6.6.6.6 activate
 neighbor 6.6.6.6 send-community extended
 neighbor 6.6.6.6 route-reflector-client
exit-address-family
```

PEs are the provider's view of the customer site, and maintain the VPN states.
In MPLS VPN, LSRs that function as ingress and/or egress routers to the VPN are often called PE (Provider Edge) routers. Devices that function only as transit routers are similarly called P (Provider) routers. Those core devices provide the transport across the SP backbone. In MPLS VPN, PE routers participate in customer routing; provide optimum routing between sites and easy provisioning of sites. PE routers allow customers to use overlapping addresses and contain a separate set of routes for each customer; isolation between routers.

BGP is used to set the path for MPLS packet forwarding across different AS, exchanges eBGP routing information between PE and CE, and uses iBGP between PE and PE in BGP/MPLS VPN. PE router receives IP prefix from CE and add 8 bytes RD (Route Distinguisher) to the IP prefix . VPN is established with defining VRF that is consisted of RD and route target (RT).

PE to CE Routing Sessions

In traditional environments, customer networks prefer to use BGP in their networks and as a PE-CE routing protocol
In an MPLS VPN network, BGP attributes for a VPN site are transparently transported across the backbone to another site in the same VPN- consistent end-to-end routing policy. Because there is a single routing protocol used across the VPN between service provider core and customer sites, the concept of redistribution does not apply. MP-BGP is generally used across the SP network
BGP PE-CE peering in an MPLS VPN environment can be performed in two different ways:
BGP PE-CE VPN sites implementing unique AS numbers
BGP PE-CE VPN sites implementing same AS numbers

Plan: place R1 and R2, and R6 and R7 (BGP between them) into two VRFs, then advertising the two over the VPNv4 network
This is done by putting "VPN-A" on both R6 and R2 for the tunnel

R2
show ip vrf VPN-A --- starting off we just see VPN-A from before RD 2.2.2.2:2 on Fa0/0
show ip route vrf VPN-A ---shows 12.12.12.0 directly connected, Fa0/0
show ip bgp vpnv4 all --- nothing here yet
debug ip bgp vpnv4 unicast updates
!
---- Here we don't do anything under the global BGP- it is all under address-family. That is- we aren't dealing with the global routing table but rather, the specific VRF table

```
router bgp 1
 address-family ipv4 vrf VPN-A
  neighbor 12.12.12.1 remote 2
  neighbor 12.12.12.1 activate
```

--- as soon as these are done, debugging is going to show routes populating from R1 and fill the screen
NOTE: **un all** will turn off all debugging

show ip vrf VPN-A
show ip route vrf VPN-A
show ip bgp vpnv4 all

Below we can see the BGP route is now coming from over the VRF, and the we can see R1 over VRF in the sh bgp output

```
     11.0.0.0/25 is subnetted, 2 subnets
B       11.11.11.0 [20/0] via 12.12.12.1, 00:00:31
B       11.11.11.128 [20/0] via 12.12.12.1, 00:00:31
     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/0
NetStepByStep-R2#show ip bgp vpnv4 all
BGP table version is 3, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 2.2.2.2:2 (default for vrf VPN-A)
*> 11.11.11.0/25    12.12.12.1               0             0 2 i
*> 11.11.11.128/25  12.12.12.1               0             0 2 i
```

On R6, What we just put in was delivered over the PE-PE network
(R2 learned from R1, advertised prefixes to R4, which passed it to R6 via the PE-to-PE routing exchange of VPNv4 prefixes)

show ip vrf VPN-A
show ip route vrf VPN-A
show ip bgp vpnv4 all

```
     67.0.0.0/24 is subnetted, 1 subnets
C       67.67.67.0 is directly connected, FastEthernet1/0
     11.0.0.0/25 is subnetted, 2 subnets
B       11.11.11.0 [200/0] via 2.2.2.2, 00:01:04
B       11.11.11.128 [200/0] via 2.2.2.2, 00:01:04
NetStepByStep-R6#show ip bgp vpnv4 all
BGP table version is 5, local router ID is 6.6.6.6
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 2.2.2.2:2
*>i11.11.11.0/25    2.2.2.2                  0    100      0 2 i
*>i11.11.11.128/25  2.2.2.2                  0    100      0 2 i
Route Distinguisher: 6.6.6.6:6 (default for vrf VPN-A)
*>i11.11.11.0/25    2.2.2.2                  0    100      0 2 i
*>i11.11.11.128/25  2.2.2.2                  0    100      0 2 i
```

So now, R6 needs to advertise just like R2 did, so the VRF will be tied together
!
conf t
router bgp 1
 address-family ipv4 vrf VPN-A
  neighbor 67.67.67.7 remote 3
  neighbor 67.67.67.7 activate

show ip vrf VPN-A
show ip route vrf VPN-A
show ip bgp vpnv4 all
--- here we can now see R7 showing up in the VRF

```
     67.0.0.0/24 is subnetted, 1 subnets
C       67.67.67.0 is directly connected, FastEthernet1/0
     77.0.0.0/24 is subnetted, 1 subnets
B       77.77.77.0 [20/0] via 67.67.67.7, 00:00:13
     11.0.0.0/25 is subnetted, 2 subnets
B       11.11.11.0 [200/0] via 2.2.2.2, 00:02:14
B       11.11.11.128 [200/0] via 2.2.2.2, 00:02:16
NetStepByStep-R6#show ip bgp vpnv4 all
BGP table version is 6, local router ID is 6.6.6.6
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 2.2.2.2:2
*>i11.11.11.0/25    2.2.2.2                  0    100      0 2 i
*>i11.11.11.128/25  2.2.2.2                  0    100      0 2 i
Route Distinguisher: 6.6.6.6:6 (default for vrf VPN-A)
*>i11.11.11.0/25    2.2.2.2                  0    100      0 2 i
*>i11.11.11.128/25  2.2.2.2                  0    100      0 2 i
*> 77.77.77.0/24    67.67.67.7               0             0 3 i
```

R1
show ip route
ping 77.77.77.77 source 11.11.11.11

R7
show ip route
ping 11.11.11.131 source 77.77.77.77

And "!!!!!" 100% successful

PE to CE when CE is running RIP
Customers who have deployed RIP as their intra-site routing protocol
This means preferred usage of RIP as the VPN inter-site routing protocol in an MPLS VPN environment.
When RIP is employed all routing information learned from a VPN site is placed in the associated VRF instance
P routers that attach to the VPN use BGP to internally distribute
Other end CE router can then learn the routes through that end's PE

R1 on the CE is just running RIP. We also need to disable BGP on it for this (remove the neighbor)
conf t
router rip

```
 no auto-summary
 version 2
 network 12.0.0.0
 network 11.0.0.0

router bgp 2
 no neighbor 12.12.12.2 remote-as 1   ---removed!
```

R2

```
show ip vrf VPN-A
show ip route vrf VPN-A
show ip bgp vpnv4 all
!
debug ip bgp vpnv4 unicast updates
!
conf t
router rip
 address-family ipv4 vrf VPN-A     ----- here is how you put the VRF in
  no auto-summary
  version 2
  network 12.12.12.0
   redistribute bgp 1 metric 2         ----- even though this is inside the VRF table, it is said this lets VRF know about BGP
router bgp 1
 address-family ipv4 vrf VPN-A
  no neighbor 12.12.12.1 remote 2     ---- similar to before (on R1) this is going to get rid of the BGP neighbor
  redistribute rip                              ---- and this tells it to refer to RIP for the routing info inside the VRF (inside VRF BGP?)
----------------RIP routes redistributed to BGP inside the VRF- is this because the example keeps BGP on R7 (CE destination)??
----------------See below

show ip vrf VPN-A
show ip route vrf VPN-A              ----- recall that this is what is INSIDE the VRF
show ip bgp vpnv4 all              ----- And note that this is querying the BGP table
```

Here are RIP prefixes from R1 along with what R6 is currently advertising
I am not sure why it is such a great idea to have R1 showing up in the BGP table- especially if it is a "tunnel"
We also did not configure VRF on R1 in this example.

```
     77.0.0.0/24 is subnetted, 1 subnets
B       77.77.77.0 [200/0] via 6.6.6.6, 00:23:21
     11.0.0.0/25 is subnetted, 2 subnets
R       11.11.11.0 [120/1] via 12.12.12.1, 00:00:17, FastEthernet0/0
R       11.11.11.128 [120/1] via 12.12.12.1, 00:00:17, FastEthernet0/0
     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/0
NetStepByStep-R2#show ip bgp vpnv4 all
BGP table version is 13, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 2.2.2.2:2 (default for vrf VPN-A)
*> 11.11.11.0/25    12.12.12.1               1         32768 ?
*> 11.11.11.128/25  12.12.12.1               1         32768 ?
*> 12.12.12.0/24    0.0.0.0                  0         32768 ?
*>i77.77.77.0/24    6.6.6.6                  0    100      0 3 i
Route Distinguisher: 6.6.6.6:6
*>i77.77.77.0/24    6.6.6.6                  0    100      0 3 i
```

R1 ---- the example didn't do anything except ping.... also didn't bother setting up RIP on the R7 end!
This could by the reason why RIP routes were redistributed to BGP inside the VRF on R2 (?)
show ip rip database
show ip route
ping 77.77.77.77 source 11.11.11.11

R7
show ip route
ping 11.11.11.131 source 77.77.77.77

PE to CE when CE is running OSPF
Confirmed this is running BGP on R7-R6 and simply adding OSPF to R1-R2

R1
conf t
no router rip
router ospf 1
 router-id 1.1.1.1
 network 1.1.1.1 0.0.0.0 area 0
 network 11.11.11.11 0.0.0.0 area 0
 network 11.11.11.131 0.0.0.0 area 0
 network 12.12.12.1 0.0.0.0 area 0

R2
show ip vrf VPN-A
show ip route vrf VPN-A

```
show ip bgp vpnv4 all
!
debug ip bgp vpnv4 unicast updates
!
conf t
no router rip

!
router ospf 12 vrf VPN-A             ---- specify VRF table after OSPF process ID
 network 12.12.12.2 0.0.0.0 area 0
 redistribute bgp 1 subnets       ----- same as with RIP.  Note the use of "subnets" here

router bgp 1
 address-family ipv4 vrf VPN-A
  no redistribute rip
  redistribute ospf 12 vrf VPN-A        ------ with RIP we didn't have to specify process ID,
but also didn't have to specify VRF

show ip vrf VPN-A
show ip route vrf VPN-A
show ip bgp vpnv4 all

R1
show ip route
ping 77.77.77.77 source 11.11.11.11

R7
show ip route
ping 11.11.11.131 source 77.77.77.77
```

PE to CE when CE is running static routes
Pretty much the same as before:

```
R1
conf t
 no router ospf 1
ip route 0.0.0.0 0.0.0.0 12.12.12.2


R2
show ip vrf VPN-A
show ip route vrf VPN-A
show ip bgp vpnv4 all
!
debug ip bgp vpnv4 unicast updates
!
conf t
```

no router ospf 12 vrf VPN-A
ip route vrf VPN-A 11.0.0.0 255.0.0.0 12.12.12.1 <----------how to add a static route in a VRF

router bgp 1
 address-family ipv4 vrf VPN-A
  no redistribute ospf 12 vrf VPN-A
  redistribute static <----------

show ip vrf VPN-A
show ip route vrf VPN-A
show ip bgp vpnv4 all

InterAS VPN

R1 - first clean up old clutter
conf t
 no ip route 0.0.0.0 0.0.0.0 12.12.12.2
router bgp 2
no network 11.11.11.0 mask 255.255.255.128
no network 11.11.11.128 mask 255.255.255.128

R2 - first clean up old clutter
conf t
no ip route vrf VPN-A 11.0.0.0 255.0.0.0 12.12.12.1
!
interface Ethernet0/3
 no ip vrf forwarding VPN-A
 ip address 12.12.12.2 255.255.255.0
!
router bgp 1
 address-family ipv4 vrf VPN-A
  no redistribute static

Now we can set things up:

THIS IS SETTING UP THE TUNNEL ON THE CE:
R1 in AS2
conf t
router bgp 2
 neighbor 12.12.12.2 remote-as 1 <----------R2 in AS1
 no bgp default route-target filter
---- overrides default behavior for whether or not you have the prefix "because you may or may not have the VRF on your router"
 !
 address-family vpnv4
 neighbor 12.12.12.2 activate

---- activating peer (R2) to exchange routes

R2 in AS1
router bgp 1
 no bgp default route-target filter   <---- just like before
 neighbor 12.12.12.1 remote-as 2   <------------R1 in AS2
 !
 address-family vpnv4
 neighbor 12.12.12.1 activate
 neighbor 4.4.4.4 next-hop-self
----- If you need to pass on those VPNv4 routes that you have exchanged over eBGP, you need to set next-hop-self for VPNv4 addresses

show ip bgp vpnv4 all
----------In this one, the video ran the above command on R2 and it did not reveal anything the narration said it did.

More on no bgp default route-target filter
Use to disable automatic BGP route-target community filtering.
If you configure the router for BGP route-target community filtering, all received eBGP VPN-IPv4 routes are discarded when those routes do not contain a route-target community value that matches the import list of any configured VRFs.
This is the desired behavior for a router configured as a PE router.

AS formal definition: a single network or group of networks that is controlled by a common system administration group and that uses a single, clearly defined routing protocol.)
Some VPNs need to extend across multiple service providers (overlapping VPNs). Regardless of complexity or location, connection must be seamless to the customer.

The inter-AS for MPLS VPNs feature provides that seamless integration of AS and service providers.
Separate AS from different service providers can communicate by exchanging IPv4 network layer reachability information (NLRI) in the form of VPN-IPv4 addresses. The AS' border edge routers use Exterior Border Gateway Protocol (eBGP) to exchange that information. Then, an interior gateway protocol (IGP) distributes the network layer information for VPN-IPv4 prefixes throughout each VPN and each AS.

Routing information uses the following protocols:
Within an AS, routing information is shared using an IGP.
Between AS, routing information is shared using an eBGP. eBGP allows a service provider to set up an interdomain routing system that guarantees the loop-free exchange of routing information between separate AS.

Configuration
An MPLS VPN with inter-AS support allows a service provider to provide to customers scalable Layer 3 VPN services, such as web hosting, application hosting, interactive learning, electronic commerce, and telephony service. A VPN service provider supplies a secure, IP-based network that shares resources on one or more physical networks.

The primary function of eBGP is to exchange network reachability information between AS, including information about the list of AS routes. The AS use eGBP border edge routers to distribute the routes, which include label switching information. Each border edge router rewrites the next-hop and MPLS labels.

Inter-AS configurations supported in an MPLS VPN can include:

Interprovider VPN—MPLS VPNs that include two or more AS, connected by separate border edge routers. The AS exchange routes using EBGP. No interior gateway protocol (IGP) or routing information is exchanged between the AS.

BGP Confederations—MPLS VPNs that divide a single AS into multiple sub-AS, and classify them as a single, designated confederation. The network recognizes the confederation as a single AS. The peers in the different AS communicate over EBGP sessions; however, they can exchange route information as if they were IBGP peers.

The inter-AS MPLS VPN feature provides the following benefits:
Allows a VPN to cross more than one service provider backbone.
The inter-AS for MPLS VPNs feature allows service providers, running separate AS, to jointly offer MPLS VPN services to the same end customer. A VPN can begin at one customer site and traverse different VPN service provider backbones before arriving at another site of the same customer. Previous MPLS VPN could only traverse a single BGP AS service provider backbone. The inter-AS feature allows multiple AS to form a continuous (and seamless) network between customer sites of a service provider.

Allows a VPN to exist in different areas.
The inter-AS for MPLS VPNs feature allows a service provider to create a VPN in different geographic areas. Having all VPN traffic flow through one point (between the areas) allows for better rate control of network traffic between the areas.
Allows confederations to optimize iBGP meshing. The inter-AS MPLS VPNs feature can make iBGP meshing in an AS more organized and manageable. You can divide an AS into multiple, separate sub- AS and then classify them into a single confederation (even though the entire VPN backbone appears as a single AS). This capability allows a service provider to offer MPLS VPNs across the confederation because it supports the exchange of labeled VPN-IPv4 NLRI between the sub-AS that form the confederation.

L2VPN:

MPLS L2 VPN

MPLS L2VPN has two modes: Virtual Private LAN Service (VPLS) and Virtual Leased Line (VLL).In the industry, a Virtual Leased Line is also referred to as Virtual Private Wire Service (VPWS)

Virtual Leased Line (VLL)
Virtual Leased Line (VLL) is a way to provide Ethernet-based point to point communication over IP/MPLS networks.VLL uses the pseudo-wire encapsulation for transporting Ethernet traffic over an MPLS tunnel across an IP/MPLS backbone.

Pseudowire

A pseudowire (PW) is an emulation of a native service over a packet switched network (PSN). The native service may be ATM, frame relay, Ethernet, low-rate TDM, or SONET/SDH, while the PSN may be MPLS, IP (either IPv4 or IPv6), or L2TPv3.

Point-to-Point (VPWS/VLL/Pseudowire):
- Virtual Leased Line (VLL) in Circuit Cross Connect (CCC) Mode
- Virtual Leased Line (VLL) in Martini Mode (PWE3) aka EoMPLS aka Xconnect
- Virtual Leased Line (VLL) in Kompella Mode

Martini vs Kompella (VLL)
Martini VLL (Virtual Leased Line) – this is a method of providing one point to point L2 link between two endpoints in the MPLS network by using LDP as a signaling protocol to transfer tunnel identification.

Kompella VLL (Virtual Leased Line) – this is exactly the same L2 point-to-point service as previous Martini VLL has, the difference is this one uses BGP as a signaling protocol to transfer tunnel identificaiton.

Point to Multipoint/Multipoint to Multipoint:
- Virtual Private LAN Service (VPLS) => A Layer-2 service that emulates a switched Ethernet (V)LAN across a PSN.

Martini vs Kompella (VPLS)
Martini VPLS (Virtual Private LAN Service) – in this service, you create an illusion that the entire MPLS cloud is a giant switch for the customer, the “Martini” again means using LDP as signaling protocol.

Kompella VPLS (Virtual Private LAN Service) – in this service you again create an illusion of a giant switch to the customer, but internally it will use BGP for signalling.

Other L2VPN types:
- 802.1q Tunneling (QinQ)

- E-VPN http://blogs.cisco.com/tag/e-vpn/).
- Frame Relay (Old) - Point to point
- ATM (Old) - Point to point

L3VPN:
- IPSEC - Point to Point
- GRE - Point to Point
- MPLS/BGP L3 VPN
- DMVPN

Implementing Cisco Service Provider Next Generation Edge Network Services (642-889)

Exam Description: The 642-889 SPEDGE Implementing Cisco Service Provider Next-Generation Edge Network Services exam is associated with the CCNP® Service Provider certification. This 90-minute, 65–75 questions exam tests a candidate's knowledge on the concepts and implementation of VPN solutions from the Service Providers perspective, including simple and complex layer 3 MPLS VPNs, CSC, 6VPE, and layer 2 VPNs such as AToM and VPLS. This exam covers the Cisco IOS, IOS-XE and IOS-XR operating systems. Candidates can prepare for this exam by taking the Implementing Cisco Service Provider Next Generation Edge Network Services (SPEDGE) course. The exam is closed book and no outside reference materials are allowed.

The following topics are general guidelines for the content likely to be included on the exam. However, other related topics may also appear on any specific delivery of the exam. In order to better reflect the contents of the exam and for clarity purposes, the guidelines below may change at any time without notice.

15%	**1.0**	**VPN in Service Provider IP NGN Environments**
	1.1	Describe VPN implementation models (overlay, peer-to-peer)
	1.2	Describe VPN technologies (L2TPv3, GRE, IPsec VPN, SSLVPN, DMVPN, GETVPN)
	1.3	Describe L2 vs L3 VPNs
40%	**2.0**	**MPLS layer 3 VPNs in Service Provider IP NGN Environments**
	2.1	Describe MPLS L3 VPN architecture and operations (RDs, RTs, VRFs, MP-BGP, PE-CE routing)
	2.2	Describe the design models for combining Internet access with MPLS L3 VPN services
	2.3	Describe the various methods used to deploy IPv6 over MPLS (6PE and 6VPE)
	2.4	Implement MP-BGP between PE routers on IOS-XR and IOS-XE
	2.5	Implement PE-CE routings (static, EIGRP, OSPF, BGP) on IOS-XR and IOS-XE
	2.6	Implement complex MPLS layer 3 VPNs on IOS-XR and IOS-XE
	2.7	Implement carrier supporting carrier (CSC) on IOS-XR and IOS-XE
	2.8	Troubleshoot MPLS L3 VPNs IOS-XR and IOS-XE configuration errors in service provider environments
19%	**3.0**	**Layer 2 VPNs in Service Provider IP NGN Environments**
	3.1	Describe L2TPv3 VPNs over an IP core network
	3.2	Describe L2 VPNs (AToM and VPLS) over an IP/MPLS core network
	3.3	Describe AToM Interworking
	3.4	Implement AToM on IOS-XR and IOS-XE
16%	**4.0**	**Carrier Ethernet in Service Provider IP NGN Environments**
	4.1	Describe Carrier Ethernet forums and standards (MEF, IEEE, IETF)

4.2	Describe the concepts of User PE (U-PE) and Network PE (N-PE)
4.3	Describe E-Line versus E-LAN versus E-Tree
4.4	Describe QinQ tunneling
4.5	Describe Provider Backbone Bridge (PBB - aka MAC-in-MAC)
4.6	Describe VPWS versus VPLS
4.7	Describe VPLS versus H-VPLS
4.8	Describe VPLS signaling using LDP or BGP
4.9	Implement QinQ on Cisco ME 3400 Series Switches
4.10	Implement VPLS on IOS-XR and IOS-XE

http://www.techexams.net/forums/ccie/99372-l2-vpns-vs-l3vpns.html